



**UNIVERSIDAD
NACIONAL
DE LOJA**



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

ESCUELA DE INGENIERÍA EN SISTEMAS

Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.

Tesis, previa a la obtención del título de Ingeniero en Sistemas.

Autor:

Lady del Cisne Iñiguez Carchi.

Director:

Ing. Boris Marcel Díaz Pauta, MBA.

**LOJA-ECUADOR
2017**

Certificación del Director

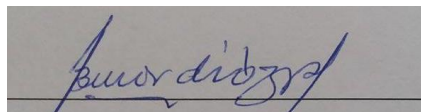
Ing. Boris Marcel Díaz Pauta, MBA.

DOCENTE DE LA ESCUELA DE INGENIERÍA EN SISTEMAS

CERTIFICA:

Que la Egresada **Lady del Cisne Iñiguez Carchi**, ha realizado bajo mi tutoría el presente Trabajo de Tesis denominado **“DESARROLLO DE UNA PROPUESTA DE MEJORA DE PROCESOS EN EL ÁREA DE ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE BASADA EN EL MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI v1.3) PARA LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN DE LA UNIVERSIDAD NACIONAL DE LOJA”**, el mismo que ha sido orientado y discutido bajo mi asesoramiento y cumple con la reglamentación pertinente; razones por las cuales reúne la suficiente validez técnica y práctica, por consiguiente autorizo su certificación para su posterior presentación y sustentación.

Loja, 29 de Marzo del 2017



Ing. Boris Marcel Díaz Pauta, MBA.

DIRECTOR DE TRABAJO DE TITULACIÓN.

Autoría

Yo: **LADY DEL CISNE IÑIGUEZ CARCHI** declaro ser autora del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el repositorio institucional – Biblioteca Virtual.

Firma:

A rectangular box containing a handwritten signature in blue ink. The signature is stylized and appears to be the name of the author, Lady del Cisne Ñiguez Carchi.

Cédula: 1105333908

Fecha: Loja, 29 de Marzo del 2017

Carta de Autorización

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo: **LADY DEL CISNE IÑIGUEZ CARCHI**, declaro ser autora de la tesis titulada: **“DESARROLLO DE UNA PROPUESTA DE MEJORA DE PROCESOS EN EL ÁREA DE ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE BASADA EN EL MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI v1.3) PARA LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN DE LA UNIVERSIDAD NACIONAL DE LOJA”**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se puede responsabilizar por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veinte días del mes de junio del dos mil diecisiete.

Firma:



Autor: Lady del Cisne Iñiguez Carchi

Cédula: 1105333908

Dirección: Peñón del Oeste

Correo Electrónico: ldiniguez@unl.edu.ec

Celular: 0982778952

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Boris Marcel Díaz Pauta, MBA.

Tribunal de Grado: Ing. Luis Roberto Jácome Galarza, Mg. Sc.

Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Ing. Carlos Miguel Jaramillo Castro, Mg. Sc.

Dedicatoria

El presente trabajo de tesis, lo dedico de manera muy especial a mis queridos padres, Jorge Arturo Iñiguez Ordoñez, Zoila Enid Carchi Vaez, quienes con sus sabios consejos, amor, esfuerzo y apoyo incondicional han sido el pilar fundamental de mi vida.

A mis hermanos Germán, Jorge, Hartman, Edgar, Hernán, Hugo y Patricio, a mis sobrinos y a toda mi familia por su cariño y su inmenso amor que me han sabido guiar por el camino de la sabiduría y de esta manera poder terminar mis estudios con gran éxito, ya que brindándome su apoyo incondicional, su confianza, he podido alcanzar todos mis ideales, y a concluir con éxito esta etapa de mi vida.

Lady del Cisne Iñiguez Carchi

Agradecimiento

Con especial aprecio y consideración, agradezco la colaboración de todos quienes aportaron con la información necesaria para la elaboración de esta Tesis, en especial al Ing. Hernán Leonardo Torres Carrión Mg. Sc, por su valiosa y acertada dirección en la contribución al desarrollo de la misma, aportando con sus valiosas sugerencias para poder llevar adelante el trabajo de investigación.

Agradezco a la Ing. Lissette López quien labora en la Unidad de Telecomunicaciones de la Universidad Nacional de Loja por su apoyo al proceso de desarrollo del presente trabajo y en general a todos mis familiares y amigos que me ayudaron moral y espiritualmente.

A todos muchas gracias.

La Autora:

Lady del Cisne Iñiguez Carchi

Tabla de Contenidos

Certificación del Director	II
Autoría	III
Carta de Autorización	IV
Dedicatoria	V
Agradecimiento	VI
1. Título	1
2. Resumen	2
2.1 Summary	3
3. Introducción	4
4. Revisión de Literatura	6
4.1. FASE INICIO	6
4.1.1. Modelos y Normas de mejora de procesos.....	6
4.1.2. Mejora de procesos software (SPI).....	12
4.1.3. Proceso software.....	13
4.1.4. Proceso.....	13
4.1.5. Metodologías, modelo y estrategia de mejora de procesos.....	14
4.2. FASE DIAGNÓSTICO	18
4.2.1. Análisis FODA.....	18
4.2.2. Línea Base.....	18
4.3. FASE ESTABLECIMIENTO	19
4.3.1. Estándar BPMN.....	19
4.3.2. Herramientas para pruebas de software.....	20
4.4. FASE ACTUACIÓN Y APRENDER	30
4.4.1. Manual de Instalación de herramientas.....	30
4.4.2. Mejora continua.....	34
5. Materiales y Métodos	36
5.1. Materiales.....	36
5.2. Métodos de Investigación.....	37
5.3. Técnicas de recolección de datos.....	38
6. Resultados	39
6.1. FASE INICIO: Obtención del Patrocinio por parte del GDS.....	40

6.1.1.	Introducción	40
6.1.2.	Patrocinio	43
6.1.3.	Descripción GDS	43
6.1.4.	Estructura Organizacional.....	43
6.1.5.	Definición del equipo de trabajo en al área PPQA	44
6.2.	FASE DIAGNÓSTICO: Análisis de la situación actual del GDS.....	47
6.2.1.	Introducción	47
6.2.2.	Situación actual del GDS.....	47
6.2.3.	Estructura funcional del GDS	48
6.2.4.	Matriz FODA del GDS - Estrategias.....	50
6.2.5.	Árbol Causa-Efecto	52
6.2.6.	Árbol de Objetivos.....	54
6.3.	FASE ESTABLECIMIENTO: Desarrollar el Plan de SQA	57
6.3.1.	Introducción	57
6.3.2.	Roles y Responsabilidades.....	57
6.3.3.	Definición de políticas para el PPQA	58
6.3.4.	Flujo de Procesos.....	59
6.3.5.	Productos de trabajo – Plantillas	74
6.3.6.	Comparativa de Herramientas	98
6.4.	FASE ACTUACIÓN - APRENDIZAJE: Validación de la propuesta.....	115
6.4.1.	Introducción	115
6.4.2.	Validación de plantillas.....	115
6.4.3.	Ejecución de pruebas	160
6.4.4.	Análisis del mejoramiento al implementar CMMI-DEV	170
	Desarrollo del Módulo “Planificación y Desarrollo Institucional”	170
	PROCESO ACTUAL vs PROCESO PROPUESTO	170
7.	Discusión.....	173
7.1.	Desarrollo de la propuesta	173
7.1.1.	Realizar un análisis de la situación actual del departamento de desarrollo de software de la Unidad de Telecomunicaciones e Información con respecto al aseguramiento de la calidad del software.....	173
7.1.2.	Desarrollo de la propuesta de políticas, estándares, procesos y procedimientos de Aseguramiento de la Calidad de Software que se adecue a las necesidades del departamento de software de la UTI.....	173
7.1.3.	Analizar herramientas a utilizar para el proceso de aseguramiento de la calidad de software y recomendar las adecuadas.	174

7.1.4. Validar la propuesta con un proyecto en ejecución en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.	174
7.2. Valoración técnica económica ambiental.....	175
8. Conclusiones	176
9. Recomendaciones	177
10. Bibliografía	178
11. Anexos	184

Índice de Figuras

Figura 1. Análisis costo/beneficio promedio CMMI e ISO 9001 [26]	11
Figura 2. Comparativa de modelos de calidad [25].....	11
Figura 3. Dimensión central de una organización [2]	13
Figura 4. Pasos de un proceso [29].....	14
Figura 5. Ciclo PDCA [30]	15
Figura 6. Ciclo de DMAIC [6]	16
Figura 7. Modelo IDEAL [33].....	17
Figura 8. Herramienta Testlink.....	30
Figura 9. Instalación de Testlink.....	30
Figura 10. Creación de la Base de datos de Testlink.....	31
Figura 11. Ingreso a Testlink.....	31
Figura 12. Pantalla Principal de Testlink.....	32
Figura 13. Comprobar la versión de Java.....	32
Figura 14. Descarga de JMeter	32
Figura 15. Carpeta bin donde se encuentra el ejecutable	33
Figura 16. Ejecutable de JMeter.....	33
Figura 17. Pantalla principal de JMeter	33
Figura 18. Mejora Continua [58].....	34
Figura 19. Organigrama proporcionado por la UTI	44
Figura 20. Árbol Causa-Efecto.....	52
Figura 21. Árbol de Objetivos	55
Figura 22. Proceso actual de Desarrollo de Software del GDS	60
Figura 23. Proceso actual de Control de Calidad del GDS.....	61
Figura 24. Proceso de Desarrollo de Software	62
Figura 25. Proceso Plan de Aseguramiento de la Calidad del Proceso y del Producto (PPQA)	66
Figura 26. Proceso Validar Documentación.....	68
Figura 27. Proceso Plan de Pruebas.....	70
Figura 28. Proceso Control de Calidad.....	72
Figura 29. Top 10 de los riesgos de seguridad en aplicaciones web. [89].....	104
Figura 30. Árbol de Casos de Prueba	160
Figura 31. Ejemplo de realización de pruebas funcionales	161

Figura 32. Matriz de resultado de pruebas v1.0.....	162
Figura 33. Matriz de resultados de pruebas v1.1.....	163
Figura 34. Documento de reporte de pruebas	163
Figura 35. Thread Group.....	165
Figura 36. HTTP Request.....	166
Figura 37. Reportes de resultados	166
Figura 38. Árbol de resultados.....	167
Figura 39. Summary Report	167
Figura 40. Tabla de resultados.....	168
Figura 41. Niveles de pruebas [68].....	210

Índice de Tablas

TABLA I. REPRESENTACIÓN DE LOS NIVELES DE CAPACIDAD DE MADUREZ [2]	6
TABLA II. REPRESENTACIÓN A NIVEL DE CAPACIDAD Y NIVEL DE PROCESO [5]	7
TABLA III. NIVELES DE SIX SIGMA.....	8
TABLA IV. CASOS DE ÉXITO EN LA IMPLEMENTACIÓN DE MODELOS Y NORMAS DE MEJORA DE PROCESOS.....	8
TABLA V. COMPARATIVA DE MODELOS Y NORMAS DE MEJORA DE PROCESOS.....	10
TABLA VI. RECURSOS HUMANOS.....	36
TABLA VII. RECURSOS TÉCNICOS Y TECNOLÓGICOS	36
TABLA VIII. RECURSOS MATERIALES.....	37
TABLA IX. APROXIMACIÓN DEL COSTO REAL DEL TRABAJO DE TITULACIÓN.....	37
TABLA X. CUADRO COMPARATIVO DE METODOLOGÍAS, MODELO Y ESTRATEGIA.....	41
TABLA XI. CASOS DE ÉXITO.....	42
TABLA XII. RESPONSABILIDADES DEL GDS.....	44
TABLA XIII. GRUPO DE SPI.....	45
TABLA XIV. ANÁLISIS FODA.....	50
TABLA XV. MATRIZ FODA.....	51
TABLA XVI. CRITERIOS DE PRIORIZACIÓN	53
TABLA XVII. CLASIFICACIÓN DE PROBLEMAS.....	53
TABLA XVIII. ESPECIFICACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE.....	63
TABLA XIX. ESPECIFICACIÓN DEL PROCESO PPQA.....	67
TABLA XX. ESPECIFICACIÓN DEL PROCESO VALIDAR DOCUMENTACIÓN.....	69
TABLA XXI. ESPECIFICACIÓN DEL PROCESO PLAN DE PRUEBAS	71
TABLA XXII. ESPECIFICACIÓN DEL PROCESO CONTROL DE CALIDAD	73
TABLA XXIII. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS FUNCIONALES.....	99
TABLA XXIV. SISTEMA OPERATIVO.....	101
TABLA XXV. SOPORTE Y DOCUMENTACIÓN.....	101
TABLA XXVI. FORMATO DE RESULTADOS.....	102
TABLA XXVII. REQUISITOS MÍNIMOS DEL MÓDULO A PROBAR.....	102

TABLA XXVIII. RESULTADOS	103
TABLA XXIX. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS DE SEGURIDAD	105
TABLA XXX. SISTEMA OPERATIVO.....	107
TABLA XXXI. SOPORTE Y DOCUMENTACIÓN.....	107
TABLA XXXII. SOPORTE DE PRUEBAS	107
TABLA XXXIII. RESULTADOS	108
TABLA XXXIV. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS DE RENDIMIENTO.....	110
TABLA XXXV. SISTEMA OPERATIVO	112
TABLA XXXVI. SOPORTE Y DOCUMENTACIÓN.....	112
TABLA XXXVII. PROTOCOLOS SOPORTADOS.....	112
TABLA XXXVIII. REQUISITOS MINIMOS DEL MÓDULO A PROBAR	113
TABLA XXXIX. RESULTADOS.....	113
TABLA XL. CASOS DE PRUEBA DE SEGURIDAD A NIVEL DE APLICACIÓN	164
TABLA XLI. RESULTADO DE PRUEBAS DE RENDIMIENTO	169
TABLA XLII. COMPARATIVA DE LAS ACTIVIDADES REALIZADAS CON EL PROCESO ACTUAL vs LA PROPUESTA.....	170
TABLA XLIII. RESOLUCIÓN DE PROBLEMAS ENCONTRADOS EN EL GDS.....	171
TABLA XLIV. COMPARATIVA DE LO QUE SE TIENE Y LO QUE SE PROPUSO.	171
TABLA XLIX. CLASIFICACIÓN DE NIVELES DE PRUEBAS	211
TABLA L. TIPOS DE PRUEBA DE SISTEMA [68]	212

1. Título

**DESARROLLO DE UNA PROPUESTA DE MEJORA DE PROCESOS EN EL
ÁREA DE ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE BASADA EN
EL MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI v1.3) PARA
LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN DE LA
UNIVERSIDAD NACIONAL DE LOJA.**

2. Resumen

El presente trabajo de titulación consiste en el desarrollo de una propuesta de mejora de procesos en el Área de Aseguramiento de la Calidad del software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) conjuntamente con el modelo IDEAL para la Unidad de Telecomunicaciones e Información (UTI) de la Universidad Nacional de Loja (UNL), el objetivo es plantear políticas, roles y responsabilidades así como también procesos y procedimientos que se adecuen al departamento de software de la UTI. Para poder validar la propuesta se tomó específicamente como piloto de pruebas el desarrollo del Módulo de Planificación y Desarrollo Institucional del “Sistema de Información Académico Administrativo Financiero” (SIAAF) de la Universidad Nacional de Loja.

Para el planteamiento de políticas, roles y responsabilidades me base en la norma propuesta por el Comité Internacional de Cualificaciones de Pruebas de Software (ISTQB), de la misma manera para el mejoramiento de los procesos de desarrollo de software se tomó como base el estándar Modelo y Notación de Procesos de Negocio (BPMN), en el levantamiento de las plantillas se revisaron varios estándares que luego de un análisis se pudo determinar los adecuados para estas instancias acoplándolos a las necesidades requeridas en este proyecto, tales como IEEE 830-1998 para la documentación o especificación de requisitos, IEEE 730-2002 con el fin de definir el plan de Aseguramiento de la Calidad del Procesos y del Producto (PPQA), el Lenguaje Unificado de Modelado (UML) para la especificación de casos de uso e ISO/IEC/IEEE 29119-3:2013 para definir el plan de pruebas.

Para realizar las pruebas al Módulo de Planificación y Desarrollo Institucional se realizó un análisis de los tipos de pruebas y de las herramientas a utilizar en base a las pruebas a realizar, determinando que las herramientas a utilizar son Testlink y JMeter.

2.1 Summary

The following qualification work consists on the development of a proposal for the process improvement in the Software Quality Assurance Area based on the Integrated Maturity Capacity Model (CMMI v1.3) jointly with the IDEAL model for Telecommunications and Information Unit (TIU) of the National University of Loja (NUL), the objective is to propose policies, roles and responsibilities as well as processes and procedures that fit to the software department from (TIU). In order to validate the proposal a pilot test was taken specifically of the Institutional Planning and Development Module of the “Financial Administrative Academic Information System” (FAAIS) of the National University of Loja.

In order to formulate policies, roles and responsibilities I based myself on the standard proposed by International Software Testing Qualifications Board (ISTQB), in the same way for the improvement of the software development processes was based on the Business Process Model and Notation (BPMN) standard, in the uprising of the templates we reviewed several standards that after an analysis could be determined suitable for these instances by matching them to the requirements required in this project, such as IEEE 830-1998 for documentation or requirements specification, IEEE 730-2002 in order to define the of Process and Product Quality Assurance (PPQA), Unified Modeling Language (UML) for specification cases of use and ISO/IEC/IEEE 29119-3:2013 to define the test plan.

To perform the tests to the Planning and Institutional Module, an analysis was made of the types of tests and tools to be used based on the test too be performed, determining that the tools to be used are Testlink and JMeter.

3. Introducción

El software es una parte vital de toda organización ya sea que esta lo produzca o lo utilice es por ello que asegurar la calidad de software es cada vez más importante por lo tanto es fundamental tener procesos que garanticen la buena calidad del producto software, permitiendo obtener un producto de alto nivel dando un valor agregado al grupo de trabajo como al departamento de desarrollo de software.

El proceso de desarrollo de software muchas de las veces no se cumplen a cabalidad no siguiendo un orden en el desarrollo del mismo, existiendo casos donde primero se desarrolla y luego se realiza la parte de documentación sin tomar en cuenta el aseguramiento de la calidad.

La propuesta de mejora de procesos en el Área de Aseguramiento de la Calidad del Proceso y del Producto cuyo piloto de pruebas fue el módulo de Planificación y Desarrollo Institucional del Sistema de Información Académico Administrativo Financiero (SIAAF), tiene la finalidad de servir como una base para el mejoramiento del desarrollo de futuros productos software en la Unidad de Telecomunicaciones e Información.

El presente trabajo de titulación fue desarrollado considerando los objetivos que se han planteado los mismos que son: realizar un análisis de la situación actual del departamento de desarrollo de software de la Unidad de Telecomunicaciones e Información con respecto al aseguramiento de la calidad del software; desarrollo de la propuesta de políticas, estándares, procesos y procedimientos de Aseguramiento de la Calidad de Software que se adecue a las necesidades del departamento de software de la UTI; analizar herramientas a utilizar para el proceso de aseguramiento de la calidad de software y recomendar las adecuadas y validación de la propuesta con un proyecto en ejecución en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.

Este proyecto se encuentra estructurado de la siguiente manera: RESUMEN contiene una breve y rápida descripción de lo que trata el trabajo de titulación, seguido tenemos la INTRODUCCIÓN donde se describe en forma general todo el ámbito del trabajo de titulación tomando en cuenta lo más relevante, luego la REVISIÓN LITERARIA la misma que permitió obtener conocimientos previos y necesarios para el desarrollo de la propuesta de mejora de procesos en el Área de Aseguramiento de la Calidad del Proceso y del Producto en base al modelo CMMI v1.3 conjuntamente con el modelo IDEAL.

En la siguiente sección los MATERIALES Y MÉTODOS se describe los métodos de investigación utilizados y técnicas de recolección de información. A continuación se presenta los RESULTADOS donde se detalla el cumplimiento de los objetivos planteados, DISCUSIÓN se especifican los objetivos con el resultado obtenido y finalmente con las CONCLUSIONES que se obtuvieron durante el desarrollo del proyecto y RECOMENDACIONES consejos que pueden ser aplicadas y servir en trabajos futuros.

4. Revisión de Literatura

4.1.FASE INICIO

4.1.1. Modelos y Normas de mejora de procesos

El principal enfoque para garantizar la calidad de productos software son los modelos o normas, los cuales se basan en estándares y procedimientos. “Un modelo de proceso es una colección estructurada de los elementos que describen características de procesos efectivos (aquellos que han demostrado por experiencia ser eficaces)”. Los modelos y normas más utilizados y conocidos para la mejora de procesos son: Capability Maturity Model (CMMI), ISO 9001:2000, ISO/IEC 15504, y Six Sigma. [1]

a) Modelo de Capacidad de Madurez Integrada (CMMI)

Es un modelo de evaluación y mejora de proceso de desarrollo de software en una organización. Se basa en evaluaciones que al ser aplicadas permiten definir las mejoras necesarias para alcanzar un nivel de capacidad o madurez. [2]

1). Estructura

CMMI se ha estructurado de forma que las organizaciones logren mejoras tanto a nivel de madurez de la empresa (representación por etapas) como en la capacidad de los procesos (representación continua).

TABLA I. REPRESENTACIÓN DE LOS NIVELES DE CAPACIDAD DE MADUREZ [2]

Nivel	Representación continua Niveles de capacidad	Representación por etapas Niveles de madurez
Nivel 0	Incompleto	
Nivel 1	Realizado	Iniciado
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		En optimización

b) ISO 9001:2000

Es un conjunto de estándares internacionales para sistemas de calidad, además diseñado para la gestión y aseguramiento de la calidad, especificando los requerimientos básicos para el desarrollo, producción, instalación a nivel de sistemas y a nivel de producto. [3]

1). Estructura

Las secciones en las que se divide ISO 9001:2000 son:

- Sistemas de Gestión de la Calidad (se especifica en los requisitos generales y requisitos de la documentación).
- Responsabilidades de la Gestión (se especifica en el compromiso con la dirección, enfoque al cliente, política de la calidad).
- Gestión de Recursos (se enfoca en la provisión de recursos, recursos humanos).
- Realización del Producto (se enfoca en la planificación de la realización del producto, procesos relacionados con los clientes, diseño y desarrollo).
- Medición, Análisis y Mejora (se basa en supervisión y medición, control de servicio no-conforme, análisis de datos). [3]

c) ISO/IEC 15504

ISO/IEC 15504 es un emergente estándar internacional de evaluación y determinación de la capacidad y mejora continua de procesos de ingeniería del software, con la filosofía de desarrollar un conjunto de medidas de capacidad estructuradas para todos los procesos del ciclo de vida y para todos los participantes. [4]

1). Estructura

El estándar ISO/IEC 15504 consiste en dos dimensiones por dimensión de procesos (se caracteriza por los propósitos de proceso y el resultado esperado del proceso) y dimensión de capacidad (consiste en un conjunto de atributos interrelacionados para gestionar un proceso y mejorar la capacidad de desempeñar un proceso).

TABLA II. REPRESENTACIÓN A NIVEL DE CAPACIDAD Y NIVEL DE PROCESO [5]

Nivel de capacidad	Atributo de Proceso (PA)
Nivel 1: Proceso Realizado	PA 1.1 Realización del proceso
Nivel 2: Proceso Gestionado	PA 2.1 Gestión de la realización PA 2.2 Gestión del producto de trabajo
Nivel 3: Proceso Establecido	PA 3.1 Definición del proceso PA 3.2 Despliegue del proceso
Nivel 4: Proceso Predecible	PA 4.1 Medición del proceso PA 4.2 Control del proceso
Nivel 5: Proceso Optimizado	PA 5.1 Innovación del proceso PA 5.2 Optimización del proceso

d) Six Sigma

Es una estrategia enfocada al cliente que basada en hechos y datos intenta alcanzar un nivel de la calidad, utiliza herramientas de gestión de calidad y estadística para reducir

la variación en los procesos y productos, y busca aumentar la satisfacción de las necesidades de los clientes y disminuir los costos de operación.

1). Estructura

Para identificar el nivel de Six Sigma de una organización se utiliza una medida de desempeño denominada Defectos por Millón de Oportunidades (DPMO) la cual puede ser usada como referente de comparación de calidad y defectos de empresas de diferentes tamaños y características. [6]

TABLA III. NIVELES DE SIX SIGMA

Nivel de sigma	Defectos por millón de oportunidades
6	3,4
5	233
4	6210
3	66807
2	308537
1	690000

TABLA IV. CASOS DE EXITO EN LA IMPLEMENTACIÓN DE MODELOS Y NORMAS DE MEJORA DE PROCESOS

Modelo	Empresa/ Institución	País	Nivel
CMMI	Novutek, S.C. [7]	México	CMMI-DEV v1.3 – Nivel 5
	Icosís Grupo Avalon [8]		CMMI-DEV v1.3 – Nivel 2
	Panel Sistemas Informáticos [9]	España	CMMI-DEV v1.3 – Nivel 3
	COBISCORP [10]	Ecuador	CMMI-DEV v1.3- Nivel 3
	KrugerCorporation [10]		CMMI-DEV v1.3 - Nivel 3
	Servicios Profesionales CIMA-E [10]		CMMI-DEV v1.3 - Nivel 3
	Tata Consultancy Services Limited [10]		CMMI-DEV y CMMI-SVC+SSD v1.3 - Nivel 5
ISO/IEC 15504	IZERTIS [11]	México, España, Perú y Colombia	Nivel 3
	ITWARE [12]	España	Nivel 3
ISO 9001	GRUPO BIMBO [13]	19 países y 3 continentes	—

		(América, Europa, Asia)	
	Equiver [14]	México	—
	APS GROUP [15]	Reino Unido	—
	AGRICULTURA COMERCIAL S.A. AGICOM [16]	Ecuador	—
	TUBASEC C.A [16]		—
	ECUATORIANA DE SAL Y PRODUCTOS QUIMICOS S.A. – ECUASAL [16]		—
	CORPORACION MUCHA MEJOR ECUADOR [16]		—
SIX SIGMA	MAURICE PHILLIPS CONTRACTS LTD [17]	Reino Unido	—
	FANAL [18]	Costa Rica	—
	Menang Corporation (MNCS) [19]	Malasia	—

TABLA V. COMPARATIVA DE MODELOS Y NORMAS DE MEJORA DE PROCESOS

	VENTAJAS	DESVENTAJAS
CMMI	<ul style="list-style-type: none"> • Permite asegurar que los procesos sean efectivos y duraderos. [4] • Guía paso a paso para la mejora, a través de niveles de madurez y capacidad. [4] • Permite realizar Aseguramiento de la Calidad del Proceso y del Producto. [2] • Tiene su propio método para la mejora de procesos (IDEAL), acoplándose a los objetivos de cada organización. [4] • Ayuda a las organizaciones a mejorar sus procesos de desarrollo y de mantenimiento, tanto para los productos como para los servicios. [20] • Control continuo de los proyectos, en cuanto a: Riegos, Recursos, Esfuerzo, Costos, Plazos. [21] • Existen 3 constelaciones CMMI para Desarrollo, Servicio y Adquisición. [2] 	<ul style="list-style-type: none"> • Puede ser difícil de entender. [4] • Requiere mayor inversión económica para implementarlo completamente. [4] • El proceso para lograr un nivel de madurez es largo y costoso, lo que puede significar varios años de esfuerzo. [8] • Puede ser considerado prescriptivo. [4] • El proceso de evaluación es más costoso en tiempo y esfuerzo. [21]
ISO 9001:2002	<ul style="list-style-type: none"> • Se acopla a los requerimientos de la organización sin importar su tipo, tamaño y producto suministrado. [3] • Permite realizar Aseguramiento de la Calidad del Producto. [3] • Libertad de implementación y de interpretación de los requisitos. [8] 	<ul style="list-style-type: none"> • Es muy general, no proporciona información de cómo aplicarlo a empresas de menor tamaño, tampoco proporciona directrices para su implementación en varias industrias. [4] • Costo alto al implantarlo. [4] • Si se desea obtener un rango más amplio en los objetivos de un sistema de gestión de la calidad se debe relacionar con la norma ISO 9004. [3] • Para integrar un sistema de gestión de la calidad propio de una empresa se debe alinear con la norma ISO 14001:1996 [3]
ISO/IEC 15504	<ul style="list-style-type: none"> • Mejora a nivel de gestión interna y mejor satisfacción al cliente. [4] • Permite asegurar una mejora mediante niveles de madurez. [5] • Primer modelo de procesos de dos dimensiones independientes para los procesos y la capacidad. [21] 	<ul style="list-style-type: none"> • Tiene un alto grado de dificultad para entender. [4] • Hace uso de algunas de las prácticas definidas en la norma ISO/IEC 12207. [5] • El costo de implantación es más costoso que otros modelos. [4] • Se considera inapropiado aplicar dicha certificación en organizaciones con menos de veinte (20) personas. [5]
SIX SIGMA	<ul style="list-style-type: none"> • Es un programa experimentado con éxito y centrado en los resultados económicos, que transforma a las empresas en las mejores de su clase. [22] • Produce en las empresas resultados económicos muy notables. [22] • Su objetivo es disminuir los costes de mala calidad centrándose en los resultados económicos. [22] • Cuenta con herramientas de software que hacen las estadísticas cada vez más fáciles. [22] • Tiene sus propias metodologías DMAIC y DFSS. [23] 	<ul style="list-style-type: none"> • Esta más enmarcada a organizaciones que ofrecen servicios. [23] • Tiene un alto grado de dificultad para entender. [24] • Se debe tener conocimientos generales de investigación de operaciones, arquitectura de sistemas e ingeniería concurrente. [24] • Debe existir capacitación acerca de la utilización del modelo, invirtiendo tiempo y recursos. [23]

De acuerdo a trabajos realizado [25] [26] [27] las certificaciones en los últimos años se han centrado en la aplicación de dos normas internacionales: el Modelo CMMI, y la Norma Internacional ISO 9001.

En la Figura 1, se puede observar que CMMI aunque es más costoso que la ISO 9001 demanda de menos horas de esfuerzo y tiene un mejor aumento en la productividad y calidad.

Criterio	CMMI	ISO 9001
Horas sin pérdida o ganancia	891	4,973
Horas de entrenamiento por persona	227	64
Costo de entrenamiento por persona	\$12,668	\$8,775
Esfuerzo (horas)	22,512	53,800
Reducción de tiempo en el ciclo	2.3x	1.14x
Aumento en la productividad	2.31x	1.13x
Aumento en la calidad	4.22x	1.22x
Recuperación de inversión	6:1	4:1
Años para obtener nivel 3 del SEI	4.76	1.57

Figura 1. Análisis costo/beneficio promedio CMMI e ISO 9001 [26]

En la figura 2, se observa que la ISO 9001 aunque es más sencillo su implementación, tiene varios aspectos en contra como que es muy simple donde no se evidencia una guía paso a paso, y no consta de algún método propio para la mejora de procesos. Por su parte CMMI aunque una de sus desventajas es, que es difícil de entender y se necesita mayor inversión, consta de su propio modelo IDEAL el mismo que permite implementar CMMI de manera más comprensible, además es el modelo de mayor prestigio y no solo es para el ámbito del software sino para Adquisición y Servicios.

	ISO 9001	CMMI
Ambito de aplicación	Genérico	Software y Sistemas
En su favor	El más extendido y sencillo	El de mayor prestigio
En su contra	Simple, general, no guía paso a paso	Difícil de entender, mayor inversión, prescriptivo
Procesos	Estructura propia	Estructura propia
Validación	Encuestas satisfacción	Encuestas satisfacción y casos de estudio
Objetivo	Cumplimiento de requisitos de calidad por procesos	Mejora del proceso, determinación capacidad contratista
Representación	Plana	Continua y por etapas
Técnicas análisis	Guías y listas de comprobación	Cuestionarios de evaluación
Método para mejora de procesos	Ninguno, guía ISO 9004	IDEAL, mapa guiado

Figura 2. Comparativa de modelos de calidad [25]

De acuerdo al análisis realizado en la TABLA I y en la Figura 1 y 2, de los distintos modelos y normas de mejora de procesos se evidencia que la ISO 9001 y CMMI son los modelos que están más allegados hacia una mejora de procesos específicamente al aseguramiento de la calidad del software, pero dentro de las desventajas de la ISO 9001 es, que no proporciona suficiente información de cómo aplicarlo a empresas de menor tamaño, además si se desea obtener un rango más amplio en los objetivos de un sistema de gestión de calidad se debe alinear con ciertos parámetros de la ISO 9004 y la ISO 14001, transformándola en una norma demasiado extensa y complicada es por ello que se llega a la conclusión que **el modelo más representativo e idóneo para implementar una mejora de procesos para el Aseguramiento de la Calidad del Software dentro de una organización es CMMI**, a pesar que es un modelo complejo de entender y costoso al igual que la ISO/IEC 15504 permite acoplarse a los objetivos de cada organización con el uso de su modelo IDEAL, asimismo da la opción de que cada organización decida qué área o áreas mejorar brindando la oportunidad de llevar a la organización hacia una certificación SCAMPI, dando mayor prestigio a la misma y a su producto.

4.1.2. Mejora de procesos software (SPI)

La calidad del software cada vez cobra mayor interés en organizaciones dedicadas al desarrollo de producto software siendo una preocupación a la que se le dedican muchos esfuerzos, sin embargo el software casi nunca es perfecto. [28]

Todo proyecto tiene como objetivo producir el software de la mejor calidad posible y que cumpla las expectativas de los usuarios.

El Software Engineering Institute (SEI) ha identificado varias dimensiones en las que una organización puede centrarse para mejorar su actividad por medio de los procesos utilizados en la misma. Las tres dimensiones donde normalmente se centran las organizaciones son: las personas, los métodos y procedimientos, y el equipamiento y herramientas. [2]

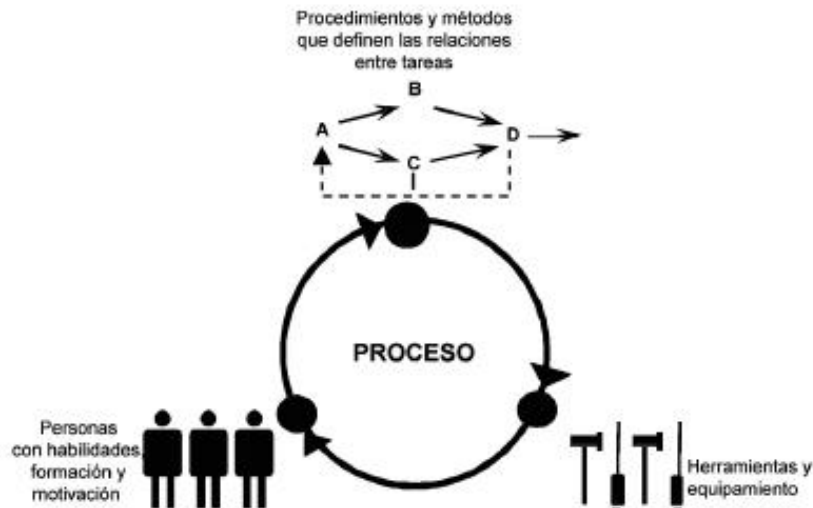


Figura 3. Dimensión central de una organización [2]

4.1.3. Proceso software

La actividad de desarrollo de software también denominado ciclo de vida es una descripción de secuencias de procesos que deben ser guiados por un equipo de individuos para generar un conjunto de productos/servicios.

Este proceso es netamente intelectual, puesto que depende mucho de la creatividad de los individuos involucrados. No existe una guía explícita para el desarrollo de software que beneficie a todo proyecto es por ello que se debe acoplar a las necesidades y requerimientos de la empresa. Proceso software es un conjunto de actividades, métodos, prácticas usadas para desarrollar y mantener software y sus productos asociados. [2]

4.1.4. Proceso

Un proceso es una secuencia de pasos con algún tipo de lógica que se enfoca en lograr algún resultado específico, los procesos son mecanismos de comportamiento que han sido diseñados para mejorar la productividad de algo, para establecer un orden o eliminar algún tipo de problema. Es importante en este sentido hacer hincapié que los procesos son ante todo procedimientos diseñados para servicio del hombre en alguna medida, como una forma determinada de accionar.

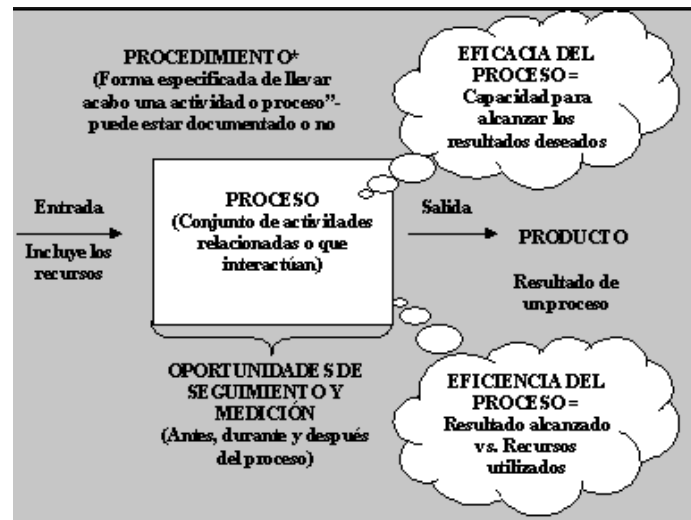


Figura 4. Pasos de un proceso [29]

4.1.5. Metodologías, modelo y estrategia de mejora de procesos

CMMI para Desarrollo (CMMI-DEV v1.3) consta de buenas prácticas que tratan las actividades de desarrollo aplicadas a productos y servicios. Aborda las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento. [2]

CMMI tiene componentes para una organización o grupo pequeño de proyectos que quieran mejorar sus procesos implantando metas, prácticas específicas y genéricas; pero no especifica como implantar por ello se requiere el uso de un modelo o metodología la cual nos indicará los pasos a seguir para adoptar CMMI.

Por tal motivo a continuación se hará un análisis preliminar de metodologías, norma y estrategia para determinar cuál es la idónea para trabajar con CMMI.

- **PDCA (Plan-Do-Check-Act) de Deming**

La estrategia de mejora continua PDCA está constituido por cuatro fases: “Planificar”, “Desarrollar”, “Verificar” y “Actuar”. [30] El mismo se utiliza para la mejora continua y lograr la resolución de problemas.

El ciclo PDCA también es denominado círculo de mejora continua, puesto que tras la última etapa comienza de nuevo la fase de planificación esta vez con valiosa información sobre los resultados obtenidos en la anterior planificación. Sus fases son:

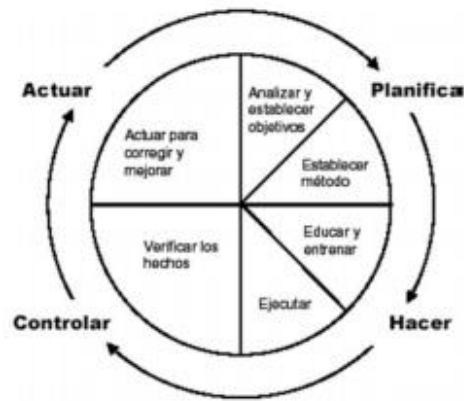


Figura 5. Ciclo PDCA [30]

- **Planificar:** En esta fase se determina los objetivos y métodos apropiados para cumplir dichos objetivos de acuerdo a los requisitos del cliente y políticas de la organización, así también como conocer la situación de la organización, la identificación de las causas raíces y la definición de soluciones potenciales a dichas causas.
- **Desarrollar:** En esta fase se orienta a las personas involucradas dentro de la organización de las actividades que se han de llevar a cabo también se realiza la ejecución del plan con las acciones previstas en la primera fase, siendo importante comenzar de manera experimental para que una vez que se ha comprobado su eficacia se conseguirá una implantación eficaz.
- **Verificar:** Aquí se realiza el seguimiento y medición de los procesos y productos respecto a las políticas, objetivos y requisitos antes previstos en los proyectos pilotos. Se ha de comprobar si los objetivos marcados se han logrado, o si no es así se comienza de nuevo desde la fase “Planificar”.
- **Actuar:** Esta es la última fase una vez que se comprueba que las acciones emprendidas dan el resultado esperado se estandarizar la nueva situación mediante una documentación adecuada describiendo lo aprendido es decir los cambios son incorporados en los procesos o actividades.

- **DMAIC o DFSS**

La primera, conocida por DMAIC se usa para mejorar los procesos existentes en una organización. La segunda conocida por DFSS se usa cuando se quiere diseñar un nuevo proceso e introducirlo en una organización de forma que las dos metodologías den soporte a las técnicas de gestión del modelo Six Sigma creado por MOTOROLA. [31]

- **DMAIC**

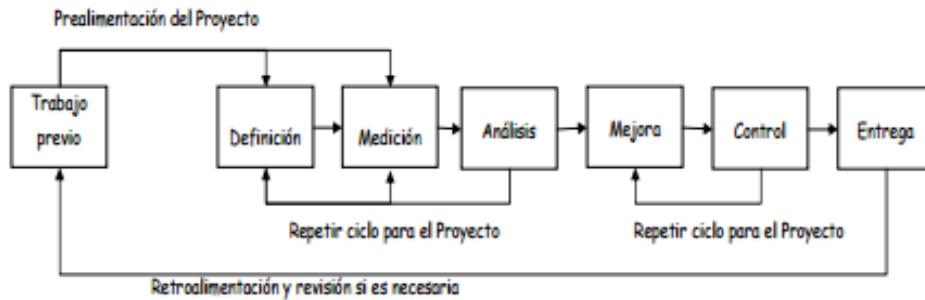


Figura 6. Ciclo de DMAIC [6]

Definir: En esta etapa se define el problema de calidad mediante una planeación que involucre las expectativas y necesidades de los clientes, la identificación del proceso y de sus interrelaciones así como también las variables críticas.

Es decir en esta primera fase se realiza un diagnostico preliminar de la organización para conocer e identificar las áreas de mejora definiendo metas, objetivos y alcance del proyecto.

Medir: La organización debe planificar e implementar procedimientos de seguimiento con el propósito de validar la información que se toma del proceso como la medición y evaluación del producto, la capacidad del proceso, los indicadores de gestión del proyecto y la satisfacción de los clientes externos e internos.

Analizar: Esta etapa es la más importante puesto que se trata de comprender las variables claves que se identificaron mediante las fases anteriores y por qué se producen defectos, y plantear que razones múltiples son identificadas como causantes. El equipo entonces busca más datos y otras evidencias para ver si encaja con la causa sospechosa. El ciclo de análisis continúa refinando la hipótesis o rechazándola hasta que la verdadera causa raíz sea identificada y verificada con los datos.

Mejorar: En esta etapa se trata de mejorar el proceso para eliminar los defectos, gracias a la etapa anterior se pueden obtener las causas de cada uno de los defectos y cuáles son los que más afectan al proceso.

Controlar: Esta etapa consiste en documentar los controles necesarios para verificar la efectividad y la eficacia de los diversos cambios que sufre el proceso. Es indispensable entonces definir indicadores que muestren el nivel de desempeño de la organización en muchos casos se hace uso de herramientas para realizar el control a un proceso, cuando se han logrado los objetivos y la misión se dé por finalizada el equipo informa a la dirección y se disuelve.

- **DFSS o DMADV**

Se le hace referencia como DMADV. Esta metodología tiene también cinco pasos:

Definir: Es la etapa inicial es donde se define el problema, los objetivos y el alcance del proyecto para llegar al propósito definido identificando el equipo de trabajo durante este proceso.

Medir: En esta etapa se comprende a fondo las necesidades de los clientes y especificaciones.

Analizar: En esta fase se asegura que todos los componentes del equipo del proyecto van en la misma dirección al crear un producto/servicio, sin problemas se trataran de cerrar todas las posibles dudas que se pueda generar alrededor del mismo.

Diseñar: Se realiza un diseño detallado del modelo detectando posibles fallos y dándoles solución consiguiendo crear un producto sin errores.

Verificar: En esta última fase se realiza la prueba piloto comunicando a los interesados de la organización como se llevará la implantación.

- **IDEAL**

EL SEI (Software Engineering Institute) ha propuesto un ciclo de mejora de procesos conocido como IDEAL. Es un modelo de mejora de procesos de software que puede ser usado como guía para iniciar, planificar e implementar acciones propensas a mejorar los procesos. Integra planes para iniciar y administrar un programa de SPI (Software Process Improvement) a través de descripciones genéricas recomendadas. [32]

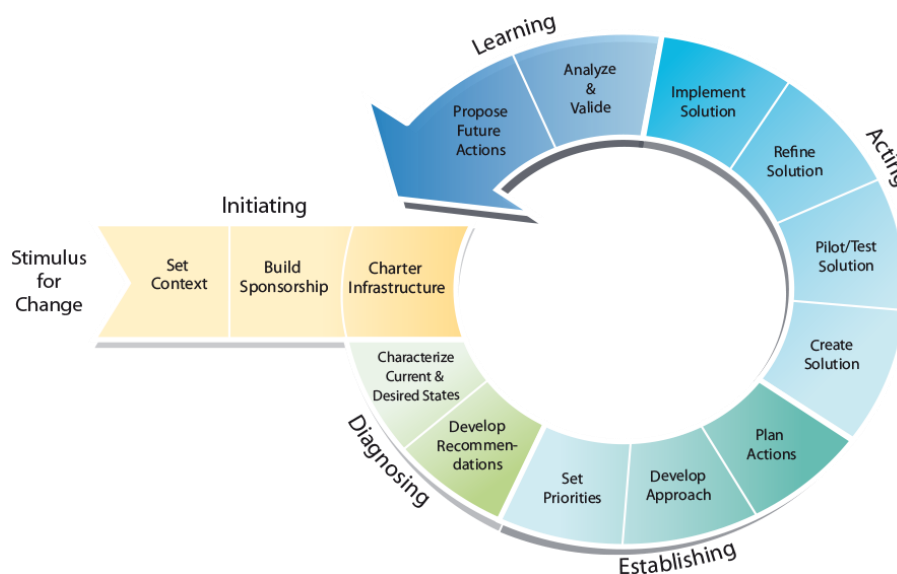


Figura 7. Modelo IDEAL [33]

- **Inicio (Initiating):** Su propósito es garantizar que el mejoramiento de procesos vaya enfocada a un factor éxito, aclarando con la gerencia los objetivos del desarrollo de la propuesta y el alcance de la misma definiendo elementos a ser evaluados y el rol y responsabilidades de cada participante.
- **Diagnóstico (Diagnosing):** Su propósito es realizar un análisis actual de la organización desarrollando recomendaciones y documentando los resultados de la fase.
- **Establecimiento (Establishing):** Su propósito es definir cuáles serán las mejoras y las recomendaciones que se realizarán definiendo por medio de un documento o plan de capacitación.
- **Actuación (Acting):** Se hace frente a las mejoras descubiertas durante el diagnóstico, se desarrollan planes para ejecutar pilotos para probar y evaluar los procesos nuevos o mejorados. Luego se prepara la adopción de toda la organización y los planes para llevar a cabo la puesta en marcha y ejecutarlo.
- **Aprender (Leveraging):** Es pasar por el modelo ideal más eficaz desarrollando soluciones y tomando en cuenta las lecciones aprendidas, estos artefactos se añaden al proceso de base de datos que se convertirá en una fuente de información para el personal involucrado en la siguiente pasada a través del modelo.

4.2. FASE DIAGNÓSTICO

4.2.1. Análisis FODA

“El análisis FODA es una herramienta que permite conformar un cuadro de la situación actual de la empresa u organización, permitiendo de esta manera obtener un diagnóstico preciso que permita en función de ello tomar decisiones acordes con los objetivos y políticas formuladas.” [34]

El análisis FODA se conforma de las primeras letras de las palabras Fortalezas, Oportunidades, Debilidades y Amenazas; las mismas que permitirán deducir las estrategias en base a estos cuatro puntos de vista.

4.2.2. Línea Base.

La línea base es un conjunto de especificaciones o productos de trabajo que se ha revisado y acordado formalmente, que sirve como la base para el desarrollo posterior y que solamente puede cambiarse mediante procedimientos de control de cambios. [2] Se ha establecido una línea base que incluye el ciclo de vida de un producto software:

1. Etapa Funcional

- Entrevistas con el cliente.

- Recopilación de requisitos.
 - Requisitos del software.
 - Requisitos de interfaz de usuario.
- Negociación de requisitos con el cliente.
 - Lista de requisitos por cambiar/modificar.
 - Requisitos modificados.
- Validación de requisitos con el cliente.
- Planificar proyecto
- Diseñar Plan de PPQA
- Documentar ERS.
- Validar ERS.

2. Etapa de Definición

- Elaboración de casos de uso y prototipo.
- Validar casos de uso y prototipo.
- Elaboración de diagramas de clase.
- Elaboración del modelo conceptual.
- Diseño de base de datos.

3. Etapa de Desarrollo

- Generación de código.
- Desarrollador envía código para ser probado.
- Líder de PPQA asigna un Tester para ejecutar casos de prueba.
- Líder de PPQA certifica funcionalidad del software.
- Paso a Producción.

4. Etapa de Producto

- Manual de usuario.
- Manuales técnicos.
- Capacitación del usuario.

4.3. FASE ESTABLECIMIENTO

4.3.1. Estándar BPMN

Los flujos de procesos han sido desarrollados en base al estándar Modelo y Notación de Procesos de Negocio (BPMN). “Es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio, esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades.” [35]

4.3.2. Herramientas para pruebas de software

El uso de herramientas para el aseguramiento de la calidad de software es una parte muy fundamental en todo proyecto, básicamente el aseguramiento de la calidad (QA) “es un medio planificado y sistemático de asegurar a la gerencia que se aplican los estándares, prácticas, procedimientos y métodos definidos del proceso” [2]. Las distintas herramientas que se analizarán servirán para los tipos de pruebas definidas en el Plan de Pruebas, los casos de prueba se ejecutan al pasar el producto software al control de calidad, de acuerdo al modelo CMMI v1.3 Control de Calidad (QC) es “parte de la gestión de la calidad orientada al cumplimiento de los requisitos de la calidad” [2] es decir asegurar la calidad del producto software detectando defectos antes que el producto software pase a producción.

- **Herramientas Funcionales**

- a) Testlink**

TestLink es una herramienta Open Source que permite crear y gestionar casos de pruebas y organizarlos en planes de prueba. Estos planes permiten a los miembros del equipo ejecutar casos de test, registrar los resultados dinámicamente, generar informes, mantener la trazabilidad con los requerimientos, así como priorizar y asignar tareas. Trabaja con bases de datos como MySQL, PostgreSQL, MS-SQL, la misma que está disponible tanto para la plataforma Windows y Linux.

Es una herramienta cuyos propósitos generales son la administración de requerimientos y aún más enfocada a las pruebas, Testlink ofrece:

1. Creación de diferentes roles con distintos permisos para los integrantes de un equipo de desarrollo.
2. Creación ilimitada de carpetas en forma de árbol (llamadas requirement-specification) para una mejor organización y agrupamiento de requerimientos.
3. Creación ilimitada de carpetas en forma de árbol (llamadas test-suites) para una mejor organización y agrupamiento de los casos de prueba.
4. Relacionar los casos de prueba con los requerimientos para indicar el grado de cobertura.
5. Versionado de casos de prueba.
6. Ejecución de los casos de prueba por versión del software bajo prueba.
7. Creación de test-plans para la ejecución y control de las pruebas.
8. Integración con herramientas de gestión de incidencias como Mantis.
9. Generación de distintos tipos de reportes: listados de pruebas, requerimientos, resultados. [36]

b) Selenium

Es un conjunto de herramientas para automatizar los navegadores web a través de muchas plataformas que nos permitirá crear conjuntos de pruebas sobre aplicaciones web. Permite la creación de scripts mediante una gran variedad de lenguajes de programación como Java, Ruby, Python, etc.

Selenium tiene el apoyo de algunos de los mayores vendedores de navegadores que han tomado (o están tomando) los pasos para hacer Selenium una parte nativa de su navegador. También es la tecnología principal en innumerables herramientas de automatización de navegadores, APIs y frameworks, sus características son:

1. Crea scripts de errores rápidos para ayudar en la automatización de las pruebas de exploración.
2. Automatización de aplicaciones web para fines de pruebas pero ciertamente no se limita a eso. Las tareas de administración basadas en la web pueden (y deberían) ser automatizadas también.
3. Cuenta con soporte de documentación y soporte en línea para resolver problemas o dudas del uso de la herramienta. [37]

c) Test Complete

Test Complete Platform tiene una arquitectura flexible que hace que la creación, mantenimiento y ejecución de pruebas automatizadas a través de escritorio, web, móvil sea fácil, rápido y rentable. Algunas de las potentes características de la plataforma Test Complete que demuestran su flexibilidad y facilidad de uso incluyen:

1. Soporte para múltiples lenguajes de script.
2. Registro de pruebas automatizadas robustas sin conocer las secuencias de comandos.
3. Escribir pruebas de regresión que no fallan al cambiar la interfaz de usuario.
4. Realizar pruebas dirigidas a datos.
5. Crear complementos y extensiones personalizadas. [38]

El Visualizador Automático de Pruebas incorporado de la plataforma Test Complete captura y muestra capturas de pantalla de cada operación realizada en la aplicación probada, proporcionando así una visión general de todo el flujo de prueba. Más importante aún, en caso de que una prueba falle estas instantáneas reducen significativamente el tiempo de depuración requerido por un desarrollador.

d) Rational Functional Tester

Es una herramienta de prueba automatizada orientada a objetos de prueba como aplicaciones Windows, .Net, Java, HTML, Siebel y SAP. Rational Functional Tester permite registrar scripts fiables y robustos que pueden reproducirse para validar construcciones nuevas de una aplicación de prueba, sus características son:

1. Se ejecuta en las plataformas Windows 2000, Windows XP y Linux.
2. Está disponible en dos entornos de desarrollo integrados y en dos lenguajes de creación de scripts.
3. Functional Tester Java Scripting utiliza el lenguaje Java e IBM Rational Software Delivery Platform. La creación de scripts de Functional Tester VB.NET 2003 y VB.Net 2005 utiliza el lenguaje VB.NET y el entorno de desarrollo Microsoft Visual Studio .NET.
4. Soporte para Ajax.
5. Genera logs de error en los cuales se identifica claramente los errores encontrados en el Script. [39]

e) Microsoft Test Manager

Test Manager es la herramienta de Microsoft para administrar y gestionar nuestro trabajo de pruebas funcionales. Con Microsoft Test Manager podemos ejecutar las pruebas para cada configuración específica de tu desarrollo, permitiendo medir el progreso que se obtiene con cada ejecución y crear diferentes informes sobre cuanto queda por probar o lo que ya se ha probado, dicha herramienta permite:

1. Crear planes de pruebas, agregar conjuntos de pruebas, crear casos de prueba o diferentes configuraciones dependiendo de la necesidad; estas se utilizan para determinar qué es lo que se quiere ejecutar o probar en los diferentes casos o conjuntos de prueba.
2. Automatizar un conjunto de pruebas y asociarlo a un caso de prueba de este modo los resultados se agregan a un plan de pruebas.
3. Las pruebas realizadas por Test Manager pueden ser gestionadas en el mismo programa pero además se pueden guardar los resultados en una base de datos y generar informes. [40]

f) Unified Functional Tester

El software HPE UFT automatiza las pruebas a través de una experiencia del usuario siendo intuitivo y visual, relaciona las pruebas manuales, automatizadas y basadas en marcos en un solo IDE. Esta solución de largo alcance reduce significativamente el coste

y la complejidad del proceso de prueba funcional e impulsa continuamente la calidad, sus características son:

1. Esta herramienta simplifica las pruebas en el ciclo de vida de las aplicaciones móviles.
2. Soporta una variedad muy extensa de tecnologías: Java, Sap, Siebel, Visual Basic .Net y Oracle entre muchas otras.
3. Posee una interfaz amigable y un código de generación de scripts en Visual Basic.
4. El soporte es muy amplio y hay gran cantidad de Ingenieros de calidad para su soporte dando fiabilidad y eficiencia. Esta herramienta se instala en cualquiera de las versiones de Windows recientes. [41]

g) Wet

WET es una herramienta de pruebas de automatización web OpenSource. WET impulsa un IE Browser directamente y por lo tanto las pruebas automatizadas realizadas con WET son equivalentes a cómo un usuario manejaría las páginas web. Utilizando WET puede realizar todas las operaciones necesarias para probar aplicaciones web como hacer clic en un vínculo, introducir texto en un campo de texto, hacer clic en un botón, etc, esta herramienta ofrece:

1. Realizar varias comprobaciones como parte del proceso de prueba mediante puntos de verificación.
2. Un probador también puede agregar bibliotecas / utilizar bibliotecas existentes para lograr casi cualquier nivel de automatización de pruebas.
3. Los scripts WET que se crean escribiendo desde la base o utilizando un navegador simulado tienen la misma sintaxis, esto permite que un equipo de prueba defina su propio equilibrio entre la generación de scripts y la secuencia de comandos.
4. WET ofrece la conveniencia de grabadoras sin comprometer los requisitos de una herramienta de automatización de prueba. [42]

h) SoapUI

SoapUI es la herramienta de prueba de API de código abierto más utilizada en el mundo para APIs de SOAP y REST. SoapUI ofrece pruebas funcionales SOAP Web Services, pruebas funcionales REST API, cobertura WSDL, pruebas de aserción de mensajes y refactorización de pruebas, sus características son:

1. El código fuente está abierto para que cualquier persona pueda realizar ediciones y mejoras en el software.
2. Tiene características fáciles de usar; interfaz gráfica y de clase empresarial.

3. SoapUI permite crear y ejecutar pruebas funcionales, de regresión, de cumplimiento y de carga automatizadas con facilidad y rapidez.
4. En un solo entorno de prueba, SoapUI ofrece cobertura de la prueba completa y apoya todos los protocolos y tecnologías estándar. [43]

• Herramientas de Seguridad

a) OWASP ZAP

OWASP Zed Attack Proxy (ZAP) es una de las herramientas de seguridad gratuitas más populares del mundo y es mantenido activamente por cientos de voluntarios internacionales. Puede ayudarle a encontrar automáticamente vulnerabilidades de seguridad en sus aplicaciones web mientras desarrolla y prueba sus aplicaciones. También es una gran herramienta para pentesters experimentados para usar para pruebas de seguridad manuales. Algunas de las funcionalidades de ZAP son:

1. Escaneo automatizado, escaneo pasivo.
2. Certificados SSL dinámicos.
3. Soporte de certificados digitales de tarjetas inteligentes y clientes.
4. Compatibilidad con sockets web.
5. Soporte para una amplia gama de lenguajes de scripting.
6. Autenticación y soporte de sesión.
7. Completamente gratis (sin pagar por la versión 'Pro'). [44]

b) Burp Suite

Burp Suite es una plataforma integrada para realizar pruebas de seguridad de aplicaciones web. Sus diversas herramientas funcionan sin problemas para soportar todo el proceso de pruebas, desde el mapeo inicial y el análisis de la superficie de ataque de una aplicación, hasta la detección y explotación de vulnerabilidades de seguridad. Burp le da un control total lo que le permite combinar técnicas manuales avanzadas con la automatización de última generación para hacer su trabajo más rápido, más eficaz y más divertido. Burp Suite contiene los siguientes componentes clave:

1. Un proxy de interceptación que le permite inspeccionar y modificar el tráfico entre su navegador y la aplicación de destino.
2. Una aplicación Spider para el rastreo de contenido y funcionalidad.
3. Un avanzado escáner de aplicaciones web para automatizar la detección de numerosos tipos de vulnerabilidades.

4. Una herramienta Intruder para realizar poderosos ataques personalizados para encontrar y explotar vulnerabilidades inusuales.
5. Una herramienta Repetidor para manipular y reenviar solicitudes individuales.
6. Una herramienta Sequencer para probar la aleatoriedad de los tokens de sesión.
7. La capacidad de guardar su trabajo y reanudar el trabajo más tarde.
8. Extensibilidad que le permite escribir fácilmente sus propios complementos para realizar tareas complejas y altamente personalizadas dentro de Burp. [45]

c) Snort

Es un sistema de prevención de intrusiones de código abierto capaz de realizar análisis de tráfico en tiempo real y registro de paquetes. Con más de 4 millones de descargas y más de 500.000 usuarios registrados, es el sistema de prevención de intrusiones más implementado en el mundo, sus características son:

1. Está disponible bajo licencia GPL gratuito y funciona bajo plataformas Windows y UNIX/Linux.
2. Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos, así como actualizaciones constantes ante casos de ataques, barridos o vulnerabilidades que vayan siendo detectadas a través de los distintos boletines de seguridad.
3. Este IDS implementa un lenguaje de creación de reglas flexibles, potente y sencilla, durante su instalación ya nos provee de cientos de filtros o reglas para backdoor, ddos, finger, ftp, ataques web, CGI, escaneos Nmap.
4. Podemos ver en consola y en tiempo real qué ocurre en nuestra red. [46]

d) WireShark

Wireshark es el analizador de protocolo de red más utilizado y ampliamente utilizado en el mundo, le permite ver lo que está sucediendo en su red a nivel microscópico y es el estándar de facto en muchas empresas comerciales y sin fines de lucro como Agencias Gubernamentales e Instituciones educativas, esta herramienta ofrece:

1. Analizador de LAN potente y rica visualmente.
2. Informes profesionales y personalizables.
3. Disparadores y alertas avanzados. [47]

e) Solarwinds

Es una herramienta que facilita el uso de registro para seguridad, detección y solución de problemas. Analiza el uso del ancho de banda de la red en tiempo real para resolver

los problemas de desempeño con más de 250.000 clientes en todo el mundo desde empresas Fortune 500 hasta pequeñas empresas, sus características son:

1. Elimina las amenazas de manera más rápida con detección instantánea de actividad sospechosa y respuestas automatizadas con fines de mitigación y cumplimiento.
2. De fácil implementación proporciona correlación de eventos en la memoria, en tiempo real, respuestas activas, monitoreo de la integridad de los archivos, inteligencia de amenazas y monitoreo de dispositivos USB.
3. Generación rápida y sencilla de informes de cumplimiento para HIPAA, PCI DSS, SOX, ISO, FISMA, FERPA, NERC CIP, GLBA, GPG13 y otros.
4. Obtención de informes/reglas de cumplimiento listos para usar, filtros preconfigurados y monitoreo de la integridad de los archivos para detectar cambios en el registro y los archivos, identifica rápidamente las violaciones de la política de cumplimiento.
5. Supervisa y analiza estadísticas de rendimiento de red detalladas y en tiempo real para enrutadores, conmutadores, puntos de acceso inalámbrico, servidores y cualquier otro dispositivo con SNMP habilitado.
6. Supervisa servidores VMware y rastrea automáticamente el rendimiento de las máquinas virtuales (VM) mediante vCenter. [48] [49]

f) GFI LANguard

GFI LanGuard permite completa administración de actualizaciones de seguridad y no seguridad para sistemas operativos Microsoft, Mac OS X, las principales distribuciones Linux y aplicaciones de terceros. Además puede automatizar también la actualización de los principales navegadores web, soporta muchas populares aplicaciones de terceros tales como Apple QuickTime, Adobe Acrobat, Adobe Flash Player, Player, Mozilla Firefox, Java Runtime y muchas más. Esta herramienta incluye:

1. Acceso remoto fácil, accede a la interfaz de usuario basada en la web a través de una conexión segura (https) compatible con todos los principales navegadores para ver los paneles de control de los productos (y los datos), los estados y los informes.
2. Informes centralizados, los clientes con redes grandes pueden instalar múltiples instancias de GFI LanGuard (sitios) y una consola web que proporciona una vista centralizada y una generación de informes agregados en todas las instancias.
3. Escalabilidad mejorada, la nueva interfaz de informes basados en web ha sido probada con una base de datos de alrededor de 60k y se ha desempeñado bien bajo estas condiciones.

4. Funciones y permisos de usuario, la nueva consola web se integra con los usuarios de Active Directory para las credenciales y la autenticación lo que significa que los clientes pueden configurar el acceso de los usuarios a cada sitio a través de AD. Los usuarios pueden tener sólo lectura (auditores) o derechos completos (admins) para cada sitio.
5. Acceso simultáneo, esto permitirá a varios usuarios usar la consola web al mismo tiempo mientras permite a varios administradores de TI reportar en diferentes partes de la red en paralelo. [50]

- **Herramientas de Rendimiento**

- a) **Jmeter**

Jmeter es una herramienta de código abierto multiplataforma Linux y Windows, una aplicación Java 100% diseñado para realizar pruebas de rendimiento. Jmeter nos permite realizar simulaciones de gran carga en el servidor, red o aplicación para comprobar su "fuerza" y para analizar el rendimiento ante diferentes tipos de sobrecarga.

Las características principales de Jmeter son las siguientes:

1. Capacidad de cargar y realizar pruebas de rendimiento en muchos tipos de aplicaciones / servidor / protocolo: Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, etc.).
2. Multihilo: permite realizar testeos de forma concurrente.
3. Interfaz gráfica: permite realizar las operaciones más rápido.
4. IDE de prueba con todas las funciones que permite la grabación rápida del plan de prueba (desde navegadores o aplicaciones nativas), creación y depuración.
5. Fácil correlación mediante la capacidad de extraer datos de formatos de respuesta más populares, HTML, JSON, XML o cualquier formato textual.
6. El marco multi-threading completo permite el muestreo simultáneo por muchos hilos y el muestreo simultáneo de diferentes funciones por grupos de hilos separados. [51]

- a) **FunkLoad**

Es un probador funcional y de carga web, escrito en Python, cuyos principales casos de uso son: Pruebas funcionales de proyectos web y por tanto pruebas de regresión también y pruebas de rendimiento, al cargar la aplicación web y supervisar sus servidores, le ayuda a identificar cuellos de botella proporcionando un informe detallado de la medición del rendimiento, las principales funciones de FunkLoad son:

1. Las pruebas funcionales son scripts puros de Python usando el framework pyUnit como prueba de unidad normal. Python permite escenarios complejos para manejar aplicaciones del mundo real.
2. Realmente emula un navegador web (single-threaded) usando una webunit mejorada de Richard Jones: Soporte de cookies, Https support, http_proxy support.
3. Convierta una prueba funcional en una prueba de carga, basta con invocar el corredor de banco para identificar problemas de escalabilidad y rendimiento. [52]

b) NeoLoad

Rápido, realista y poderoso NeoLoad es una herramienta de prueba de carga y rendimiento que simula de forma realista la actividad del usuario y supervisa el comportamiento de la infraestructura para que pueda eliminar los cuellos de botella en todas sus aplicaciones web y móviles, sus características son:

1. Tiene una nueva característica de actualización de ruta de usuario que reduce el tiempo de refactorización de script en un 90%.
2. Utilizando la GUI sin scripts y una gran cantidad de funciones de automatización, permite diseñar pruebas 5-10 veces más rápido y mantener scripts en tan sólo un 10% del tiempo de diseño original.
3. Con un soporte incomparable para WebSocket, HTTP / 2, GWT, HTML 5, flujo de bits adaptable y muchas más tecnologías.
4. Ofrece una caja de herramientas de pruebas de rendimiento completo para aplicaciones web y móviles para que pueda probar completamente las tecnologías que sus aplicaciones tienen hoy. [53]

c) Gatling

Gatling es un framework de prueba de carga de código abierto basado en Scala, Akka y Netty, es necesario tener JDK8. Permite realizar simulación de carga de usuarios concurrentes contra un sistema a través de los protocolos HTTP/S, JMS o JDBC. El escenario más típico de uso es el de simular usuarios de un sistema web para poder analizar los cuellos de botella y optimizar el sistema. Funciona sobre Java con lo cual está apta para todos los Sistemas Operativos (no es suficiente con el runtime, hace falta el kit de desarrollo), sus características son:

1. Alto rendimiento.
2. Informes HTML listos para presentar.
3. Grabador de escenarios y DSL para desarrolladores.
4. Gatling se basa en tecnologías y marcos de vanguardia como Scala, Akka y Netty; alto rendimiento y eficiencia son el ADN de Gatling. [54]

d) Qtest 5.0

QTest le permite asegurarse de que el rendimiento proporcionado por sus aplicaciones de negocio satisfaga sus compromisos de calidad de servicio. QTest detecta anomalías de rendimiento y le proporciona un diagnóstico fácil de usar, se integra en todo el desarrollo de aplicaciones y en el ciclo de control de calidad. Las anomalías de funcionamiento se identifican y analizan para permitir una corrección inmediata, sus características son:

1. Permite probar una variedad de tipos de proyectos, incluyendo aquellos basados en J2EE, .Net, SAP, Siebel, Webservices, así como proyectos Web 2.0.
2. Usa tecnologías como Adobe Flash / Flex.
3. Las aplicaciones cliente / servidor, así como las aplicaciones de visualización remota de estilo Citrix, también pueden resaltarse.
4. Un enfoque basado en los negocios en la realización de sus pruebas de rendimiento:
 - ✓ Visualización gráfica de escenarios empresariales.
 - ✓ Rápida comprensión de las pruebas con la validación de sus SLAs de aplicación.
 - ✓ Seguimiento de la actividad de cada usuario virtual.
5. Detección y análisis automático de anomalías:
 - ✓ Monitorización completa de la infraestructura de hardware y software gracias a una serie de monitores de rendimiento no intrusivos.
 - ✓ Un análisis en tiempo real de métricas completas. [55]

e) Multi Mechanize

Multi-Mechanize es un framework de código abierto para pruebas de rendimiento y carga. Ejecuta scripts de Python concurrentes para generar carga (transacciones sintéticas) en un sitio o servicio remoto. Multi-Mechanize es el más utilizado para el rendimiento web y pruebas de escalabilidad pero puede ser utilizado para generar carga de trabajo contra cualquier API remota accesible desde Python, ofrece:

1. Los informes de salida de prueba se guardan como XML compatible con HTML o JMeter.
2. Ejecuta scripts de Python concurrentes para generar carga (transacciones sintéticas) en un sitio o servicio remoto. [56]

4.4. FASE ACTUACIÓN Y APRENDER

4.4.1. Manual de Instalación de herramientas

Para las pruebas de Rendimiento y Funcionales se hizo uso de las herramientas seleccionadas como son JMeter y Testlink respectivamente, en el caso de las pruebas de Seguridad se las realiza manualmente debido a que las herramientas antes analizadas no se acoplan a las necesidades requeridas en estas instancias.

- **Herramienta TestLink**

Para la instalación de la herramienta TestLink previamente se debe tener instalado el Servido Apache, base de datos MySQL o Phpmyadmin y PHP. Una alternativa es instalar el servidor WAMP o XAMPP los mismos que ya incluyen todos los requisitos necesarios.

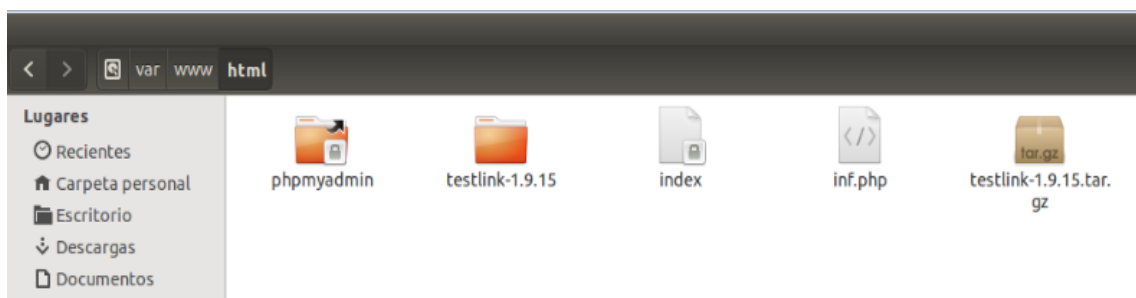


Figura 8. Herramienta Testlink

Luego de descargar Testlink de su página principal: <http://testlink.org/> copiamos la carpeta en la ruta **var/www/html** y luego se la descomprime.

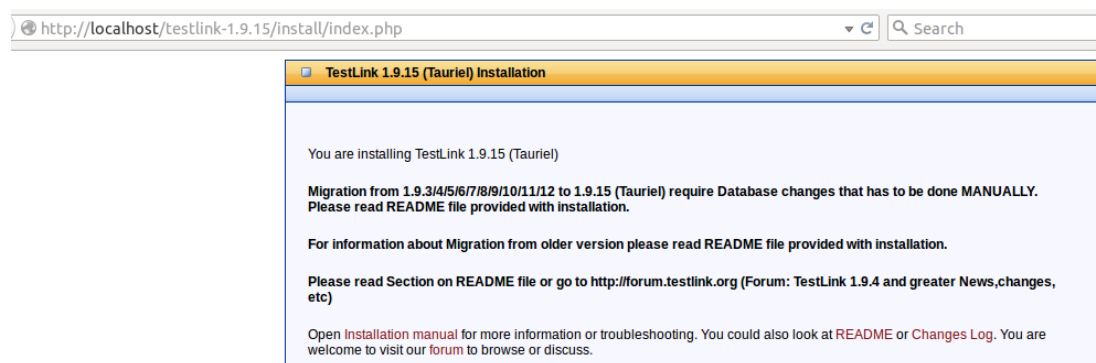


Figura 9. Instalación de Testlink

Vamos a nuestro navegador de preferencia y escribimos **localhost/testlink-1.9.15** cabe recalcar que el nombre de la carpeta depende de la versión que se descargue, luego vamos a **New Installation** y aceptamos los términos y condiciones seguido ingresamos en la carpeta de Testlink y modificamos el archivo **config.inc** en las variables

\$g_repositoryPath, **\$tlCfg->log_path** la ruta de las carpetas **upload_area** y **logs** respectivamente para evitar que de problemas en la instalación.

Set an existing database user with administrative rights (root):

Database admin login

Database admin password

This user requires permission to create databases and users on the Database Server. These values are used only for this installation procedures, and is not saved.

Define database User for Testlink access:

TestLink DB login

TestLink DB password

This user will have permission only to work on TestLink database and will be stored in TestLink. All TestLink requests to the Database will be done with this user.

After successful installation You will have the following login for TestLink Administrator:
login name: admin
password : admin

Figura 10. Creación de la Base de datos de Testlink

Seguido creamos una base de datos Testlink colocando el nombre y el password y automáticamente Testlink dará el rol de administrador con el **login name: admin** y **password: admin** presionamos **Process Testlink Setup** y ya se tendrá Testlink instalado.

TestLink 1.9.15 (Tauriel)

[Nuevo Usuario](#) [Contraseña Olvidada](#)

Figura 11. Ingreso a Testlink

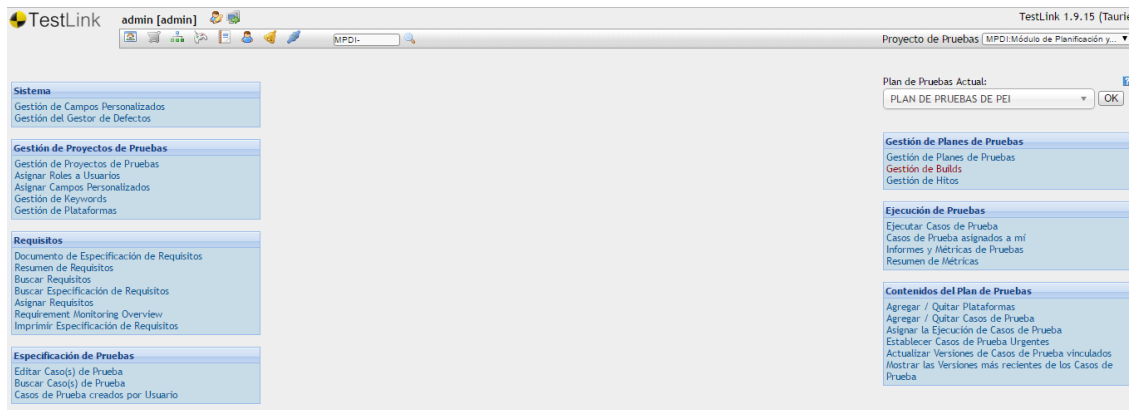


Figura 12. Pantalla Principal de Testlink

- **Herramienta JMeter**

Como requisito previo para instalar JMeter se debe tener instalado Java 7 o superior para ello nos dirigimos a la terminal y comprobamos que versión de java tenemos.

```
leidy@leidy-HP-1000-Notebook-PC: ~
leidy@leidy-HP-1000-Notebook-PC:~$ java -version
java version "1.7.0_91"
OpenJDK Runtime Environment (IcedTea 2.6.3) (7u91-2.6.3-0ubuntu0.14.04.1)
OpenJDK 64-Bit Server VM (build 24.91-b01, mixed mode)
leidy@leidy-HP-1000-Notebook-PC:~$
```

Figura 13. Comprobar la versión de Java

Visitamos la página principal de JMeter: <http://jmeter.apache.org/> y descargamos, luego se busca el archivo en Descargas y se lo descomprime.

```
leidy@leidy-HP-1000-Notebook-PC: ~/Descargas
leidy@leidy-HP-1000-Notebook-PC:~$ ls
Descargas          Documento sin titulo 1~  Imágenes          Público
Documentos        Escritorio             Música             Vídeos
Documento sin titulo 1  examples.desktop      Plantillas        workspace
leidy@leidy-HP-1000-Notebook-PC:~$ cd Descargas
leidy@leidy-HP-1000-Notebook-PC:~/Descargas$ unzip apache-jmeter-3.1.zip
```

Figura 14. Descarga de JMeter

Seguido nos dirigimos a la carpeta que se descomprimió y buscamos la carpeta **bin**, dentro de la misma se encuentra el ejecutable **ApacheJMeter.jar** el que nos permitirá acceder a la interfaz gráfica de JMeter.



Figura 15. Carpeta bin donde se encuentra el ejecutable

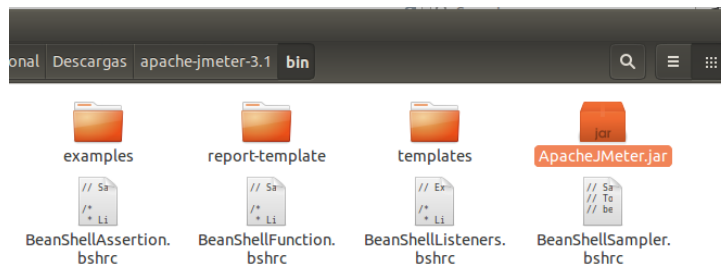


Figura 16. Ejecutable de JMeter

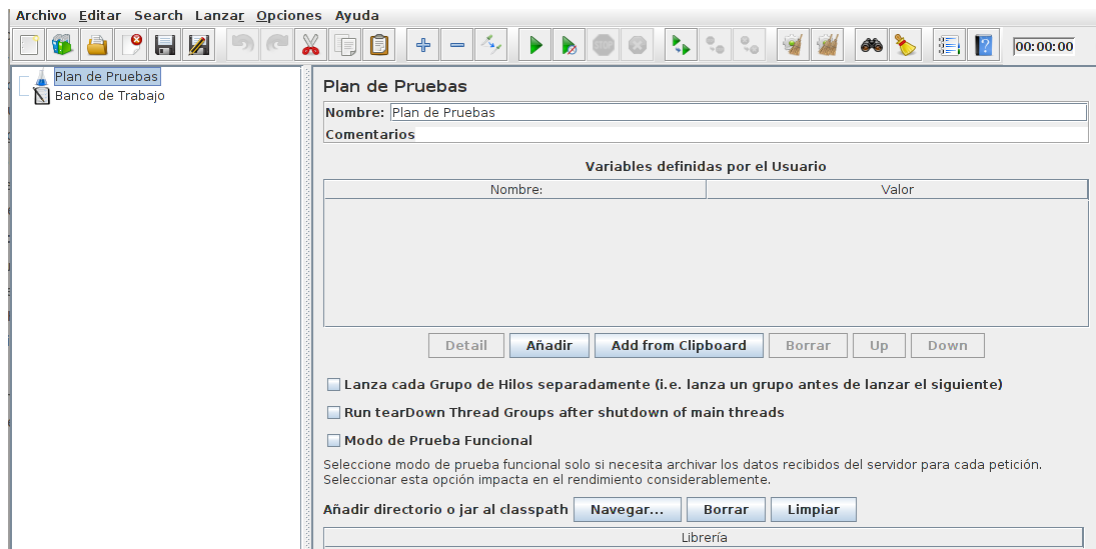


Figura 17. Pantalla principal de JMeter

4.4.2. Mejora continua

De acuerdo a la ISO [57] la mejora continua es “Mejorar la eficacia de un sistema aplicando la política de calidad, los objetivos de calidad, los resultados de las verificaciones de inspección, el análisis de los datos, las acciones correctivas y preventivas y la revisión de la Dirección.”

La mejora continua significa una mejora de la calidad de un servicio o producto el mismo que tendrá un incremento continuo y cuantificable de la satisfacción del cliente.

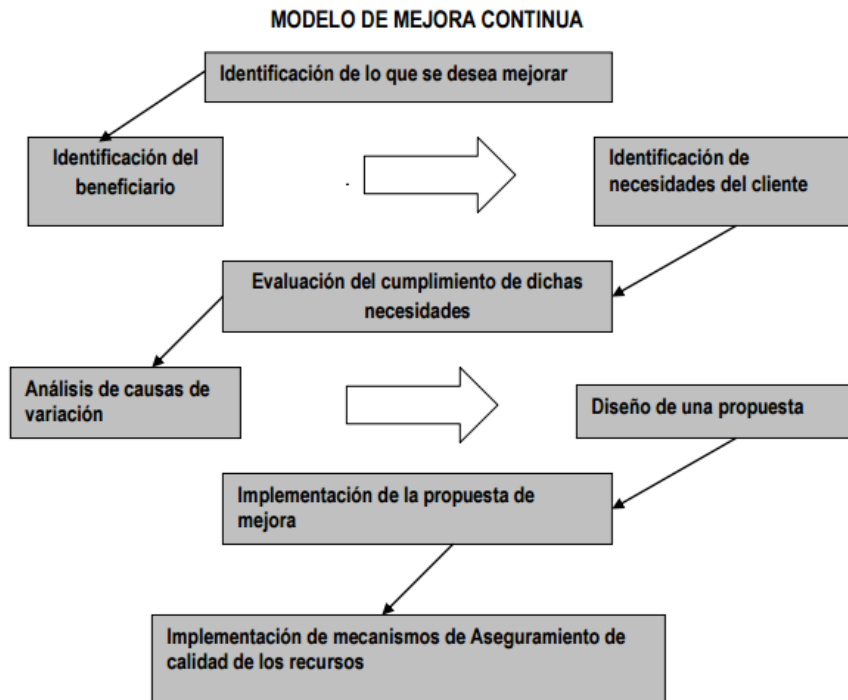


Figura 18. Mejora Continua [58]

Crear una cultura de mejora continua en una organización depende de la mentalidad de los involucrados, es decir no existen fórmulas mágicas, soluciones simples, ni decisiones rápidas para conseguirlo. El éxito está en determinar o definir un liderazgo firme que apoye la iniciativa y asigne recursos suficientes para el proyecto.

La mejor forma de afrontar los retos que conlleva la implantación de una cultura de mejora continua en la calidad es implicar a la mayor cantidad de gente posible. La forma más adecuada de conseguir una participación efectiva del personal (y como consecuencia dar pasos en la creación de una cultura de la mejora continua en la Organización), es por medio de la creación de equipos de trabajo. [59]

Los elementos para conseguir una cultura de mejora continua son: [59]

- Orientación hacia el cliente del servicio/producto (usuario).
- Liderazgo.
- Participación del personal.
- Adoptar un planteamiento de Gestión por Procesos.
- Planes de mejora continua.
- Enfoque en hechos para la toma de decisión.
- Relaciones mutuamente beneficiosas con los proveedores.

Una de las buenas prácticas para la mejora continua utilizada por organizaciones son los modelos y normas como: CMMI, ISO 15504, ISO 9001:2000 y Six Sigma. Para su adecuada implantación se requiere apoyo de la organización y retroalimentación por parte de todos los involucrados. Para tener un éxito en la implantación de los modelos antes mencionados se hace uso de metodologías, modelos o estrategias, como son: IDEAL, PDCA, DMAIC O DFSS; estas dos últimas pertenecientes al modelo Six Sigma.

5. Materiales y Métodos

Para la realización del presente trabajo de titulación, se contó con recursos humanos, económicos y tecnológicos necesarios para su culminación.

5.1. Materiales

Para la realización del trabajo de titulación fue necesario recursos humanos, técnicos- tecnológicos y recursos materiales.

- **Recursos Humanos**

TABLA VI. RECURSOS HUMANOS

Descripción	Cantidad	Horas	Costo/hora (\$)	Valor Total (\$)	Nota
Investigador	1	700	5,0	3500,0	
Director de Proyecto fin de carrera	1	400	8,0	3200,0	El costo total por hora del Director de proyecto fin de carrera y del Técnico será asumido por la Universidad.
Técnico Designado por la UTI	1	100	8,0	800,0	
TOTAL				7500,0	

- **Recursos Técnicos y Tecnológicos**

TABLA VII. RECURSOS TÉCNICOS Y TECNOLÓGICOS

HARDWARE			
Descripción	Cantidad	Valor Unitario (\$)	Valor Total (\$)
Portátil HP Pentium	1	1200,0	1200,0
Impresora	1	60,0	60,0
Internet	9(mes)	20,05	184,50
SOFTWARE			

Sistema Operativo Linux	0,0
Apache Jmeter	0,0
Testlink	0,0
TOTAL	1444,5

- **Recursos Materiales**

TABLA VIII. RECURSOS MATERIALES

Descripción	Unidad	Cantidad	Valor Unitario (\$)	Valor Total (\$)
Resma de papel A4	Hoja	1	4,80	4,80
Cartuchos de tinta	Cartucho	3	8,0	24,0
Varios				25,0
TOTAL				53.80

- **Total de recursos**

TABLA IX. APROXIMACIÓN DEL COSTO REAL DEL TRABAJO DE TITULACIÓN

Descripción	Valor Total (\$)
Recursos Humanos	7500,0
Recursos Técnicos y Tecnológicos	1444,5
Recursos Materiales	53,80
Transporte	65,0
TOTAL	9063,30

5.2. Métodos de Investigación

- **Método Inductivo**

Este método permitió establecer información pertinente y existente del departamento de desarrollo de software de la Unidad de Telecomunicaciones e Información para

determinar que inconvenientes existen actualmente en la Unidad frente al Aseguramiento de la Calidad de Software.

- **Método Deductivo**

Mediante este método se pudo obtener una solución a la mayoría de los problemas presentes mediante la propuesta de Aseguramiento de la Calidad en el desarrollo de software, satisfaciendo cada uno de los objetivos planteados en el presente proyecto.

5.3. Técnicas de recolección de datos

- **Entrevista**

Esta técnica se aplicó al Director de la Unidad de Telecomunicaciones e Información así como a todos los integrantes del GDS entre ellos el Subdirector de Desarrollo de Software y los Analistas/Desarrolladores para mayor información (**Anexo 2**), quienes conocen la actualidad del departamento de desarrollo de software lo que permitió comprender la situación actual específicamente en cuanto a las responsabilidades de cada uno de ellos, de los procesos actuales que llevan para desarrollar software y si cuentan actualmente con aseguramiento de calidad para sus productos.

- **Observación**

Con esta técnica se evidencio las actividades realizadas por los integrantes del DGS comprobando la desorganización, la poca documentación realizada al momento de desarrollar un producto software y la carga de trabajo para cada uno de los integrantes del GDS.

- **Bibliográfica**

La revisión bibliográfica se realizó de fuentes fidedignas entre la información revisada están libros, artículos científicos, tesis doctorales, tesis de magister, estándares, normas, metodologías, casos de éxito de modelos y metodologías que trabajen con CMMI.

6. Resultados

FASE 1: INICIO

Versión	Fecha	Autor
1.2	13/06/2016	Lady Iñiguez Carchi

Objetivos

Los objetivos de esta fase son:

- Profundizar los conocimientos acerca de la mejora de procesos software (SPI).
- Investigar las diferentes metodologías para la mejora de procesos.
- Establecer el patrocinio por parte del GDS de la UTI para el SPI.
- Conocer de forma general la estructura y funcionamiento del GDS de la UTI.
- Conformar el grupo de mejoramiento de procesos software referente al área PPQA.

Técnicas

- Reunión con los involucrados.
- Entrevistas.
- Análisis de material bibliográfico.
- Análisis de información brindada por el subdirector del GDS.

Involucrados

- Miembros del GDS.
- Director de Trabajo de Titulación.
- Tesista.

Problemas y Soluciones

Problema 1: Poco conocimiento de cómo adoptar el modelo IDEAL.

Solución 1: Mucha investigación y consultas a personas con conocimiento del tema.

Problema 2: Falta de tiempo de los miembros del GDS para contestar las entrevistas.

Solución 2: Insistir con las entrevistas a los miembros del GDS hasta obtener sus argumentos.

Glosario de Términos

- GDS, Grupo de Desarrollo Software.
- UTI, Unidad de Telecomunicaciones e Información.
- SDSW, Subdirección de Desarrollo Software.
- SREI, Subdirección de Redes y Equipos Informáticos.
- DBA, Administrador de Base de Datos.
- PPQA, Aseguramiento de la Calidad del Proceso y del Producto.
- IDEAL, Iniciar, Diagnosticar, Ejecutar, Aprender,
- MSG, Management Steering Group, Grupo Directivo de Gestión.
- SEPG, Software Engineering Process Group, Grupo de Procesos de Ingeniería de Software.
- CMMI, Capability Maturity Model Integration, Modelo de Capacidad de Madurez Integrada.
- SPI, Software Process Improvement, Mejora de Procesos Software.
- SEI, Software Engineering Institute, Instituto de Ingeniería Software.

6.1. FASE INICIO: Obtención del Patrocinio por parte del GDS.

6.1.1. Introducción

En esta fase se definirá el punto de partida para la Mejora de Procesos Software específicamente en el área de Aseguramiento de la Calidad del Proceso y del Producto (PPQA), empezar un proyecto de mejora no es tan fácil puesto que para continuar con el mismo se debe tener el apoyo de la alta gerencia del GDS como de sus integrantes.

Es por ello que esta etapa se empieza con reuniones con las máximas autoridades de la UTI para brindar el apoyo e información necesaria para llevar a cabo la mejora de procesos, seguido de la recolección de información bibliográfica y de una comparativa de las metodologías, modelo y estrategia más utilizados en cuanto a la mejora de procesos.

TABLA X. CUADRO COMPARATIVO DE METODOLOGÍAS, MODELO Y ESTRATEGIA

	Características	Ventajas	Desventajas
<i>IDEAL</i>	<ul style="list-style-type: none"> • Consta de 5 fases (Iniciar, Diagnosticar, Establecer, Actuar y Aprender). • Es un ciclo continuo. • El tiempo aplicado a cada iteración depende de cada organización. • El modelo IDEAL se puede aplicar a todas las áreas de proceso de una organización. • Desarrollado por el SEI para el modelo CMMI. 	<ul style="list-style-type: none"> • Garantiza que la introducción de nuevas tecnologías productos respondan a necesidades de los clientes y al deseo de mejorar continuamente. [60] • IDEAL ayuda a fomentar una cultura de mejoras continuas en la organización. [60] • IDEAL supone una gran herramienta para planificar los recursos y esfuerzos necesarios para incorporar la mejora o el cambio en la organización. [60] • Aporta un mecanismo para reunir lecciones aprendidas en cada actividad y poder acceder a ellas cuando se inicie un nuevo ciclo de mejora. [60] • IDEAL es un modelo disponible al acceso de cualquiera. 	<ul style="list-style-type: none"> • En la práctica los límites entre las fases de IDEAL no están tan claramente definidos como se muestra en el modelo. [33] • IDEAL es altamente recomendado como modelo secuencial, sin embargo en la práctica es casi imposible seguir una secuencia ya que algunas actividades se realizan en forma paralela. [61]
<i>PDCA</i>	<ul style="list-style-type: none"> • Consta de 4 fases (Planear, Desarrollar, Verificar, Actuar). • Es un ciclo continuo. • Es una metodología versátil, es decir se la puede aplicar en gran variedad de actividades. • Es muy utilizado para los Sistemas de Gestión de Calidad (SGC). • Tiene una estrecha relación con la norma ISO 9001. 	<ul style="list-style-type: none"> • Es un método efectivo y rápido para implementar la mejora continua. [62] • El ciclo PDCA se puede aplicar a todas las actividades de una organización. [62] • PDCA ayuda a implementar planes de mejora eficaces [63] 	<ul style="list-style-type: none"> • El ciclo PDCA garantiza mejoras a largo plazo. [64] • Se debe comenzar de manera experimental de tal manera que se pueda comprobar su eficacia en la fase Chequear (Check). [30] • Difícil implantación incluso para los que tienen conocimientos acerca de dicha metodología. [64] • Se requiere la implantación en toda la organización para obtener resultados exitosos. [65]
<i>DMAIC y DFSS</i>	<ul style="list-style-type: none"> • Son un ciclo continuo. • Inicialmente se las desarrolló para el sector industrial, pero hoy en día se las utiliza para diversas actividades como la mejora continua de procesos. • Desarrolladas por la empresa MOTOROLA internamente en el modelo Six Sigma. • Para alcanzar el máximo nivel se debe aplicar ciertos procedimientos de la ISO 9000. • DMAIC mejora procesos existentes en una organización y DFSS diseña nuevos procesos para introducirlos en una organización. 	<ul style="list-style-type: none"> • DMAIC y DFSS se enfocan en las necesidades y los requerimientos de los clientes. [66] • Se las puede implantar en el desarrollo de proyectos simples hasta los más complejos. • DMAIC y DFSS se requiere hacer uso de herramientas estadísticas apropiadas que conducen a soluciones válidas y efectivas. [6] 	<ul style="list-style-type: none"> • No se puede aplicar DMAIC sin antes haber aplicado DFSS. [66] • DFSS es un proceso a largo plazo y costos. [67] • DFSS es muy complicada su implantación, se debe ejecutar capacitación y bajo la supervisión de los administrativos de proyectos. [67] • Tanto para DFSS y DMAIC se debe tener conocimientos generales de investigación de operaciones, arquitectura de sistemas e ingeniería concurrente. [24]

A continuación se hace una breve reseña de casos de éxito basados en las metodologías, modelo y estrategia antes mencionadas conjuntamente con CMMI.

TABLA XI. CASOS DE ÉXITO

CASOS DE ÉXITO			
Institución	País	Metodología Utilizada	Nivel
Universidad de las Ciencias Informáticas. [68] [69]	Cuba	IDEAL	CMMI-DEV v1.3 – Nivel 2
Valores Corporations Softtek SA de CV [70]	México	DMAIC y DFSS	CMMI-DEV v1.3 – Nivel 5
Tata Consultancy Services. [71]	Asia/México/Ecuador /India/Argentina/Brazil/etc.	DMAIC y DFSS	CMMI-SVC+SSD v1.3 y CMMI-DEV v1.3 – Nivel 5
Novutek, S.C. [72]	México	IDEAL	CMMI-DEV v1.3 – Nivel 5
EMPRESAS ECUATORIANAS CERTIFICADAS CON CMMI [10]			
Institución	Ciudad	Nivel	
COBISCORP	Bogotá/Quito (sede EEUU)	CMMI-DEV v1.3(Staged):Maturity - Nivel 3	
KrugerCorporation	Quito	CMMI-DEV v1.3(Staged):Maturity - Nivel 3	
Servicios Profesionales CIMA-E	Guayaquil	CMMI-DEV v1.3(Staged):Maturity - Nivel 3	
Tata Consultancy Services Limited	Quito	CMMI-DEV v1.3(Staged):Maturity y CMMI-SVC+SSD v1.3 - Nivel 5	

Resultados del análisis

De acuerdo a la información obtenida se concluye que el modelo IDEAL cumple con las necesidades requeridas en estas instancias, IDEAL es gratuito lo cual facilita la implantación del mismo en una organización, desarrollado propiamente para CMMI por

Software Engineering Institute (SEI) dando la posibilidad de acoplarlo a los objetivos de cada organización y determinar las áreas o área de proceso a mejorar, a diferencia del modelo SIX SIGMA con las metodologías DMAIC y DFSS; así como PDCA que se requiere la implantación en toda la organización para obtener resultados exitosos. Así también destacando que existen empresas que han conseguido el éxito haciendo uso de CMMI con IDEAL y CMMI con DMAIC-DFSS alcanzando los niveles de certificación más altos otorgados por la evaluación SCAMPI, no así PDCA que es más orientada a trabajar con la ISO y no se ha podido evidenciar que existan casos de éxito hasta la actualidad al trabajar con CMMI.

De la misma manera se realizó una recopilación de las organizaciones Ecuatorianas certificadas en CMMI las mismas que cuentan con un nivel de madurez óptimo para generar productos y servicios software de calidad.

6.1.2. Patrocinio

Lo primordial en esta fase es el patrocinio por parte del departamento de desarrollo de software es por ello que se realizó una reunión con el Ing. Hernán Torres, Director y Subdirector de la UTI Ing. Milton Labanda y Ing. Edison Coronel respectivamente y la Tesista Lady Iñiguez, determinando el alcance del proyecto analizando el alcance de cada objetivo y designando técnicos para la gestión de información del GDS. El proyecto fue aprobado por las máximas autoridades del GDS como son el Director y Subdirector de la UTI, contando con el respaldo total y dando su visto bueno para el desarrollo de este proyecto para mayor información. **(Anexo 1)**

6.1.3. Descripción GDS

Para tener una idea de cómo está actualmente el GDS y los proyectos que llevan a cabo se tuvo conversaciones previas a más de la aplicación de entrevistas con los integrantes del mismo, llevando hoy por hoy varios proyectos que están estipulados en el Plan Operativo Anual (POA) pero dan prioridad a ciertos proyectos con un poco más de importancia **(Anexo 2)**.

6.1.4. Estructura Organizacional

Estructuralmente la UTI tiene su organigrama de forma general donde se especifica el orden por departamentos, específicamente haciendo hincapié al departamento de Desarrollo de Software tampoco tienen un organigrama interno definido. De tal manera que cualquiera de sus miembros puede cumplir todos los roles existentes dentro de este departamento, excepto de Subdirector de Desarrollo de Software, definiendo por cada

proyecto software existente un Líder de proyecto, Analistas, Desarrolladores y Tester el mismo que es una persona que no haya estado inmiscuido en el proyecto actual, teniendo el organigrama de la siguiente manera.

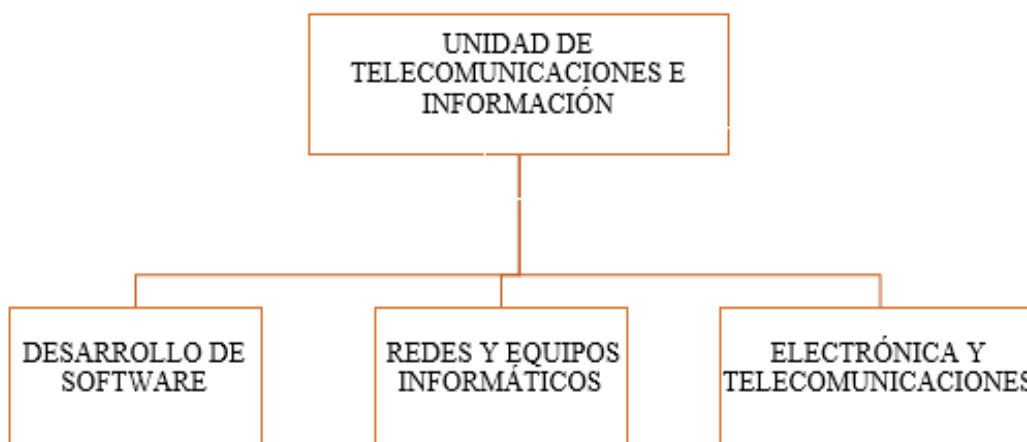


Figura 19. Organigrama proporcionado por la UTI

De acuerdo al mismo las responsabilidades que tiene cada integrante dentro del GDS son: ver **(Anexo 2)**

TABLA XII. RESPONSABILIDADES DEL GDS

	Responsabilidad
Ing. Edison Coronel	Subdirector de Desarrollo de Software
Lic. Mabel Rodríguez	Seguridad de la Información
Ing. José Martínez	Base de Datos/Desarrollador
Ing. Verónica Gonzales	Analista/Desarrollador
Ing. Jorge Malla	Analista/Desarrollador
Ing. Vladimir Romero	Analista/Desarrollador
Ing. Mario León	Analista/Desarrollador
Ing. Lissette López	Analista/Desarrollador

6.1.5. Definición del equipo de trabajo en al área PPQA

Como punto de partida para cumplir con las demás fases del modelo IDEAL es necesario definir el personal que será encargado de llevar a cabo este proyecto como Grupo de SPI, el mismo que realizará el diagnóstico, establecerá el plan de mejora y realizará las pruebas pertinentes. Determinando la o las personas que deberán estar al frente de todas estas actividades de tal manera que se pueda cumplir con las metas propuestas definiendo roles y responsabilidades de cada integrante. Para el presente proyecto se han seleccionado 2 personas las mismas que fueron asignadas por parte del Subdirector de la UTI, conformada por las siguientes personas.

TABLA XIII. GRUPO DE SPI

<i>Asesor del Proyecto</i>	Ing. Hernán Torres
<i>Asesores de SPI</i>	Ing. Edison Coronel Ing. Lissette López
<i>Equipo de trabajo</i>	Lady del Cisne Iñiguez Carchi

- **Asesor del Proyecto**

Se ha designado al Ing. Hernán Torres como Asesor del proyecto puesto que es la persona idónea al tener conocimientos en cuanto a procesos como lo plantea CMMI de la mano del modelo IDEAL, el mismo que asesora que las actividades del proyecto vayan enfocadas y de acuerdo a los lineamientos del departamento de desarrollo de software de la UTI, así también como coordinar las actividades de mejora de procesos junto al Subdirector de Desarrollo de Software de la UTI el Ing. Edison Coronel.

- **Asesores de SPI**

Se ha designado junto con el Ing. Edison Coronel Subdirector de Desarrollo de Software de la UTI las personas encargados de brindar la información necesario con respecto a cómo se lleva a cabo los procesos de software actualmente en el departamento de software, designando a la Ing. Lissette López y el mismo Ing. Edison Coronel para brindar toda la información que se requiera por la tesista.

- **Equipo de trabajo**

La Tesista Lady Iñiguez, será la responsable de coordinar, analizar y desarrollar una propuesta de mejora de procesos en el área de Aseguramiento de la Calidad del Proceso y del Producto propuesta por CMMI v1.3 en el Departamento de Desarrollo de Software de la UTI.

FASE 2: DIAGNÓSTICO

Versión	Fecha	Autor
1.4	20/07/2016	Lady Iñiguez Carchi

Objetivos

Los objetivos de esta fase son:

- Determinar el estado actual del GDS respecto al aseguramiento de la calidad del software.
- Realizar un análisis FODA para conocer el estado actual del GDS.

- Realizar un árbol causa-efecto que me permita identificar el problema central del GDS en el área PPQA.
- Realizar la categorización y prioridad de los problemas encontrados.

Técnicas

- Análisis de documentación proporcionada.
- Análisis FODA.

Involucrados

- Miembros del GDS.
- Director de Trabajo de Titulación.
- Tesista.

Problemas y Soluciones

Problema 1: Poco conocimiento para realizar la matriz FODA.

Solución 1: Investigación mediante videos tutoriales y consultas a personas con conocimiento del tema.

Problema 2: Realizar el árbol causa-efecto para obtener el problema central.

Solución 2: Basarse en el análisis FODA antes desarrollado y tener conversaciones con los integrantes del GDS.

Glosario de Términos

- UTI, Unidad de Telecomunicaciones e Información.
- FODA, Fortalezas, Oportunidades, Debilidades, Amenazas.
- GDS, Grupo de Desarrollo Software.
- MPDS, Manual de Procedimiento para el Desarrollo de Sistemas.
- TIC's, Tecnologías de la Información y la Comunicación.
- PPQA, Aseguramiento de la Calidad del Proceso y del Producto.
- UNL, Universidad Nacional de Loja.
- TIC., Tecnología de la Información y la Comunicación.
- Hw, Hardware.
- Sw, Software.
- MPDS, Manual de procedimiento para el desarrollo de sistemas.

6.2. FASE DIAGNÓSTICO: Análisis de la situación actual del GDS.

6.2.1. Introducción

Es importante conocer la situación actual del departamento de desarrollo de software para poder continuar con la mejora de procesos, de la misma manera es primordial conocer cómo se encuentra el departamento de desarrollo con respecto al Aseguramiento de la Calidad del Software.

Para llevar a cabo esta fase, se ha utilizado diferentes técnicas para la recolección de la información como entrevistas, análisis de documentación proporcionada por la UTI y análisis FODA logrando obtener información segura y verídica para el desarrollo de esta fase.

6.2.2. Situación actual del GDS.

El GDS forma parte de la UTI, a su vez esta unidad forma parte importante de Coordinación General Administrativa Financiera quien soluciona problemas internos de la Universidad.

Actualmente no se ha formalizado una misión, visión u objetivos del GDS sino más bien se rigen a un estatuto general de la UTI denominado **“LAS POLITICAS DE TELECOMUNICACIONES, DESARROLLO DE SOFTWARE, REDES” DE LA UNIVERSIDAD NACIONAL DE LOJA**. Dictaminado el 9 de Mayo de 2013 por las autoridades supremas de la UNL, donde constan varias normas y políticas para el desarrollo de software las cuales son: **(Anexo 3)**

POLÍTICAS DE LA SECCIÓN DE DESARROLLO DE SOFTWARE

- Políticas de proyectos.
- Políticas de metodologías de desarrollo.
- Políticas de demandantes de sistemas.
- Políticas de mantención de sistemas.
- Políticas de explotación de sistemas.
- Políticas de desarrollo de sistemas y aplicaciones en computadores.

POLÍTICAS, NORMAS Y PROCEDIMIENTOS PARA LA ELABORACIÓN DE SISTEMAS.

Objetivo.

Establecer las políticas, normas y procedimientos que permitan homogeneizar el desarrollo de sistemas de información dentro de la Universidad Nacional de Loja.

- Políticas para la elaboración de sistemas.
- Normas para la elaboración de sistemas.
- Normas para el análisis de sistemas.
- Normas para el diseño de sistemas.
- Normas para la programación y documentación de sistemas.
- Normas para la implantación de sistemas y capacitación.
- Normas para el mantenimiento de sistemas.
- Manual de procedimiento para el desarrollo de sistemas (MPDS).

El objetivo de estas normas y políticas es:

Definir la metodología a la que se debe someter todo el personal involucrado en la elaboración de sistemas, como objeto de obtener productos de alta calidad que resulte de fácil mantenimiento para cualquier miembro del equipo de trabajo.

De la misma manera se hace hincapié que aunque no estén formalizados los procesos de desarrollo de software que actualmente sigue el GDS para la construcción de software se basan en los mismos. Observando que actualmente se manejan procesos no enfocados al Aseguramiento de la Calidad del Proceso y del Producto. **(Anexo 4)**

6.2.3. Estructura funcional del GDS

Cabe recalcar que esta designación está estipulada en él, **MANUAL DE DESCRIPCIÓN, VALORACIÓN Y CLASIFICACIÓN DE PUESTOS DE LA UNIVERSIDAD NACIONAL DE LOJA. (Anexo 5)**

Subdirector de Desarrollo de Software

Misión

Planificar, coordinar y controlar los estudios de software necesarios para garantizar la eficacia y efectividad de los procesos académicos y administrativos de la Universidad Nacional de Loja.

Las actividades que debe cumplir son:

- Planifica, coordina, controla y evalúa el diseño, implementación y mantenimiento de software requerido por las diferentes áreas académicas o unidades administrativas de la Universidad Nacional de Loja.
- Coordina, implementa y monitorea la optimización de los sistemas informáticos de gestión académica.

- Coordina, implementa y monitorea la optimización de los sistemas informáticos de gestión administrativa y financiera.
- Coordina la preparación de las especificaciones técnicas de los pliegos para la adquisición o contratación de software especificado en base a la planificación y proyectos aprobados por la Institución.
- Establece y coordina la ejecución de las políticas de seguridad y respaldo de los sistemas informáticos de la institución.
- Coordina la capacitación a los usuarios internos de la operación del software adquirido o desarrollado por la Sección.

Analista de Sistemas Informáticos 1 y 2

Misión

Ejecutar y supervisar los procesos técnicos del sistema de Telecomunicaciones e Información de la Universidad Nacional de Loja, relacionados con desarrollo de software, diseño, implementación y mantenimiento de redes; y adquisición, renovación y mantenimiento de equipos informáticos.

Las actividades que debe cumplir son:

- Diseña, implementa y mantiene el software requerido por las diferentes áreas académicas o unidades administrativas de la Universidad Nacional de Loja.
- Ejecuta los procesos de optimización de los sistemas informáticos de gestión académica y administrativa.
- Realiza estudios, diseños, implementación y mantenimiento de redes informáticas, según los requerimientos académicos y administrativos institucionales.
- Proporciona soporte técnico y mantenimiento preventivo y correctivo a las redes y equipos de los sistemas informáticos de la Universidad Nacional de Loja.
- Elabora las especificaciones técnicas de los pliegos para la adquisición o contratación de software especializado, equipos informáticos y materiales para su mantenimiento.
- Capacita a los usuarios internos en la operación del software adquirido o desarrollado, equipos informáticos y TIC's.

Quedando una vez más al descubierto que aun en este manual no existe todos los roles y responsabilidades de los integrantes que debe tener un Grupo de Desarrollo Software.

6.2.4. Matriz FODA del GDS - Estrategias

Realizar la matriz FODA permite determinar la situación actual del departamento de desarrollo de software de la UTI y de esta manera poder proponer estrategias a los inconvenientes encontrados acorde a los objetivos del departamento, es por ello que se propone el siguiente análisis FODA del GDS de acuerdo a la información obtenida en las entrevistas realizadas al personal del mismo (**Anexo 2**).

- **FODA-GDS**

TABLA XIV. ANÁLISIS FODA

FORTALEZAS	OPORTUNIDADES	DEBILIDADES	AMENAZAS
<ul style="list-style-type: none"> • Personal con experiencia en desarrollo de software. • Apertura para la mejora de procesos software. • Apertura al cambio. 	<ul style="list-style-type: none"> • Mejorar la calidad del producto software. • Reconocimiento en el uso de herramientas tecnológicas para la mejora de proceso. • Obtener una certificación en la mejora de procesos software. • Crear una cultura de mejora de procesos software. • Tener convenios con empresas o instituciones locales. 	<ul style="list-style-type: none"> • Falta de personal para las diferentes etapas del ciclo de vida de un proyecto. • No existe una definición de roles, responsabilidades y políticas de calidad. • Escasa formalización de procesos de desarrollo software. • No contar con una persona o grupo de personas para el área de SQA. • Desconocimiento de herramientas para asegurar la calidad de un producto. • Falta de equipamiento Hw. 	<ul style="list-style-type: none"> • No se cuenta con el presupuesto necesario para emprender en una mejora de procesos. • Que al momento de optar por una certificación de mejora esta no sea otorgada. • Que de acuerdo a la figura del departamento no pueda ofrecer servicios al medio externo.

Matriz FODA-Estrategias

TABLA XV. MATRIZ FODA

<p style="text-align: center;">Factores Internos</p> <hr/> <p style="text-align: center;">Factores Externos</p>	<p>FORTALEZAS F1: Personal con experiencia en desarrollo de software. F2: Apertura para la mejora de procesos software. F3: Apertura al cambio.</p>	<p>DEBILIDADES D1: Falta de personal para las diferentes etapas del ciclo de vida de un proyecto. D2: No existe una definición de roles, responsabilidades y políticas de calidad. D3: Escasa formalización de procesos de desarrollo software. D4: No contar con una persona o grupo de personas para el área de PPQA. D5: Desconocimiento de herramientas para asegurar la calidad de un producto. D6: Falta de equipamiento Hw.</p>
<p>OPORTUNIDADES O1: Mejorar la calidad del producto software. O2: Reconocimiento en el uso de herramientas tecnológicas para la mejora de procesos. O3: Obtener una certificación en la mejora de procesos software. O4: Crear una cultura de mejora de procesos software. O5: Tener convenios con empresas o instituciones locales.</p>	<p>F1O1-2: Elaborar un plan de mejora de procesos y capacitar al personal en herramientas tecnológicas para la mejora de procesos todo esto aplicado por el personal teniendo como resultado un producto liberado de alta calidad. F2-1O3-4-5: Crear en el GDS una cultura de mejora de procesos aprovechando la apertura; proponiendo convenios con entidades conectoras del tema y poder combinar ideas desarrollando software bajo estándares básicos lo que permitirá acceder a una certificación dando más credibilidad y reconocimiento a la UNL.</p>	<p>D1-2O1: Elaborar un plan de mejora que cubra roles, responsabilidades y políticas de calidad; mejorando la calidad del producto software liberado satisfaciendo al cliente/usuario. D5O2: Capacitar a los integrantes del GDS en herramientas tecnológicas para el aseguramiento de la Calidad del Software que permita ofrecer productos con mejores prestaciones. D3O3-4: Crear una cultura y capacitar a los integrantes del GDS en la mejora de procesos, formalizando como una base los procesos de desarrollo de software partiendo de esto para lograr continuar con la mejora y tener la posibilidad de una certificación. D4O5: Firmar convenios con otras instituciones o empresas pudiendo desarrollar talleres de capacitación del personal en el área de PPQA.</p>
<p>AMENAZAS A1: No se cuenta con el presupuesto necesario para emprender en una mejora de procesos. A2: Que al momento de optar por una certificación de mejora esta no sea otorgada. A3: Que de acuerdo a la figura del departamento no pueda ofrecer servicios al medio externo.</p>	<p>F1A3: Aprovechar los conocimientos y experiencia del personal para desarrollar un producto software el mismo que podría ser ofrecido al medio externo con los más altos niveles de calidad. F3A1: Mediante la apertura por parte de la UTI al cambio en sus procesos se tiene la posibilidad que a partir de ahí se pueda conseguir el apoyo financiero por las máximas autoridades de la UNL para continuar con la mejora de procesos. F2A2: Teniendo la apertura para la mejora de procesos software y contando con el recurso humano especializado se tiene todas las posibilidades de optar por una certificación.</p>	<p>D2-3A3: Para poder ofrecer servicios al medio externo se debe primero tener un orgánico estructural y formalizar los procesos de desarrollo software para poder continuar con la mejora de procesos de una manera ordenada y eficaz. D4-6A1: Crear una cultura de mejora de procesos y formalizar un área de PPQA existiendo un plan de capacitación en el GDS en herramientas tecnológicas para asegurar que los productos liberados son de calidad.</p>

6.2.5. Árbol Causa-Efecto

Mediante el árbol causa-efecto se determina la situación actual del GDS identificando los problemas principales y categorizándolos de acuerdo a su prioridad. Tanto las causas y efectos planteados en el árbol causa-efecto son el resultado del análisis FODA, entrevistas realizadas a los miembros del GDS y de la documentación proporcionada por la UTI.

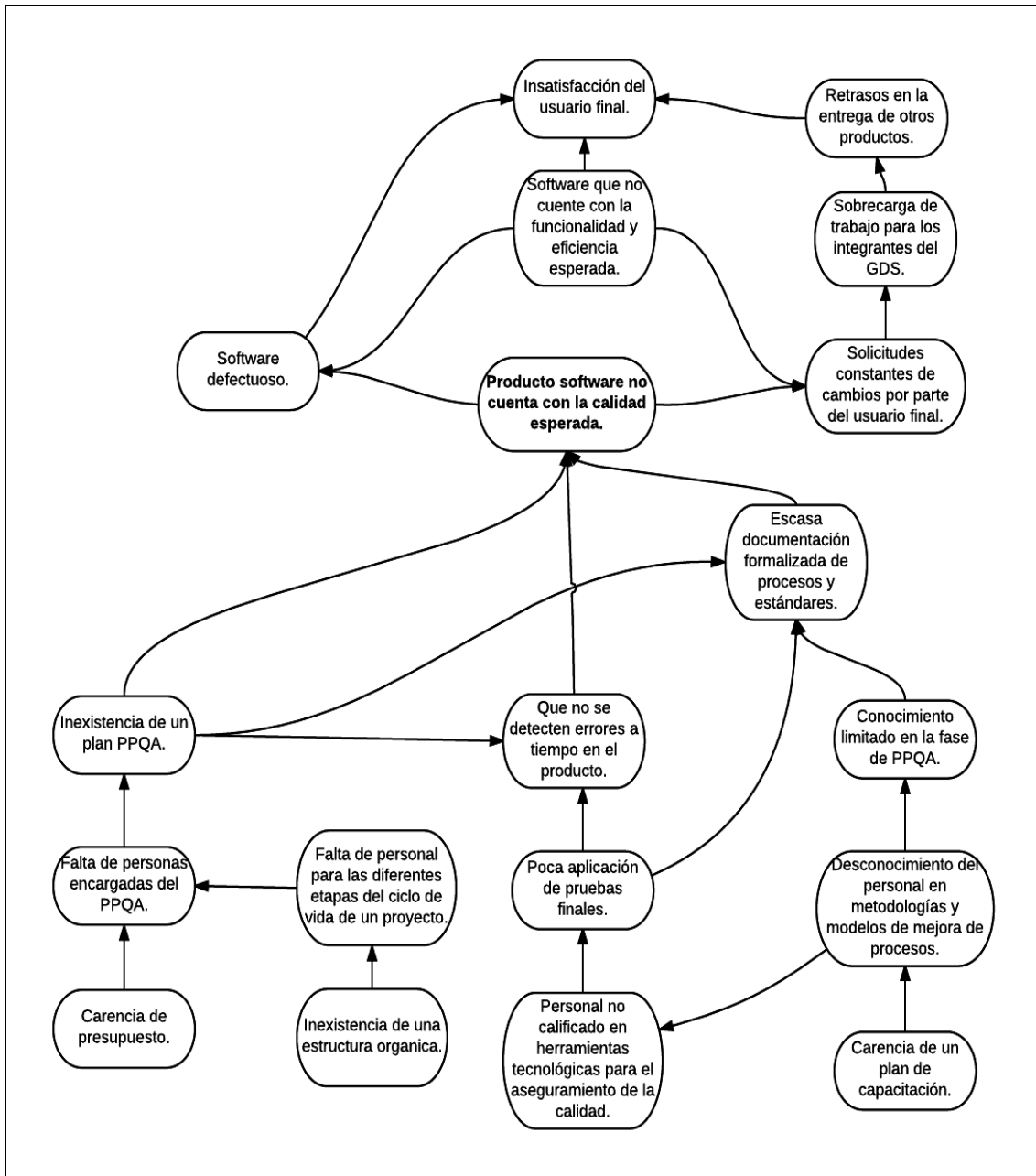


Figura 20. Árbol Causa-Efecto

Los problemas identificados en la parte inferior del árbol constituyen las causas del problema central mientras que en la parte superior se identifican los efectos ocasionados por dicho problema central.

- **Priorización**

Teniendo claro las causas y efectos del problema central se realiza una priorización de problemas. Los resultados obtenidos de esta priorización pueden utilizarse como guía para determinar qué acciones tomar en el proceso de mejora. Teniendo claro que se la realizó en base al árbol causa-efecto obtenido. Los criterios tomados para la priorización, son los siguientes:

Alto= 3

Medio= 2

TABLA XVI. CRITERIOS DE PRIORIZACIÓN

CRITERIOS DE PRIORIZACIÓN	
Prioridad (puntos)	Razón
3	Problemas que de forma directa sean parte de la solución a este proyecto o que a su vez estén dentro de mis alcances
2	Problemas que estén fuera de mi alcance es decir problemas que se generan por cuestiones externas al GDS

TABLA XVII. CLASIFICACIÓN DE PROBLEMAS

PROBLEMAS	PRIORIDAD
Escasa documentación formalizada de procesos y estándares.	3
Inexistencia de un plan PPQA.	3
Que no se detecten errores a tiempo en el producto.	2
Conocimiento limitado en la fase de PPQA.	2
Falta de personas encargadas del PPQA.	2
Falta de personal para las diferentes etapas del ciclo de vida de un proyecto.	2
Poca aplicación de pruebas finales.	2

Desconocimiento del personal en metodologías y modelos de mejora de procesos.	3
Carencia de presupuesto.	2
Inexistencia de una estructura orgánica.	2
Personal no calificado en herramientas tecnológicas para el aseguramiento de la calidad.	3
Carencia de un plan de capacitación.	2

Mediante la aplicación de las diversas técnicas como entrevistas a los integrantes del GDS, análisis FODA, árbol Causa-Efecto y priorización de problemas se considera que el problema central identificado en el GDS es que el **“PRODUCTO SOFTWARE NO CUENTA CON LA CALIDAD ESPERADA”**.

6.2.6. Árbol de Objetivos

Representa la situación esperada al resolver el problema, es decir se construye buscando las situaciones contrarias a las indicadas en el árbol causa-efecto. [73]

Árbol de Objetivos

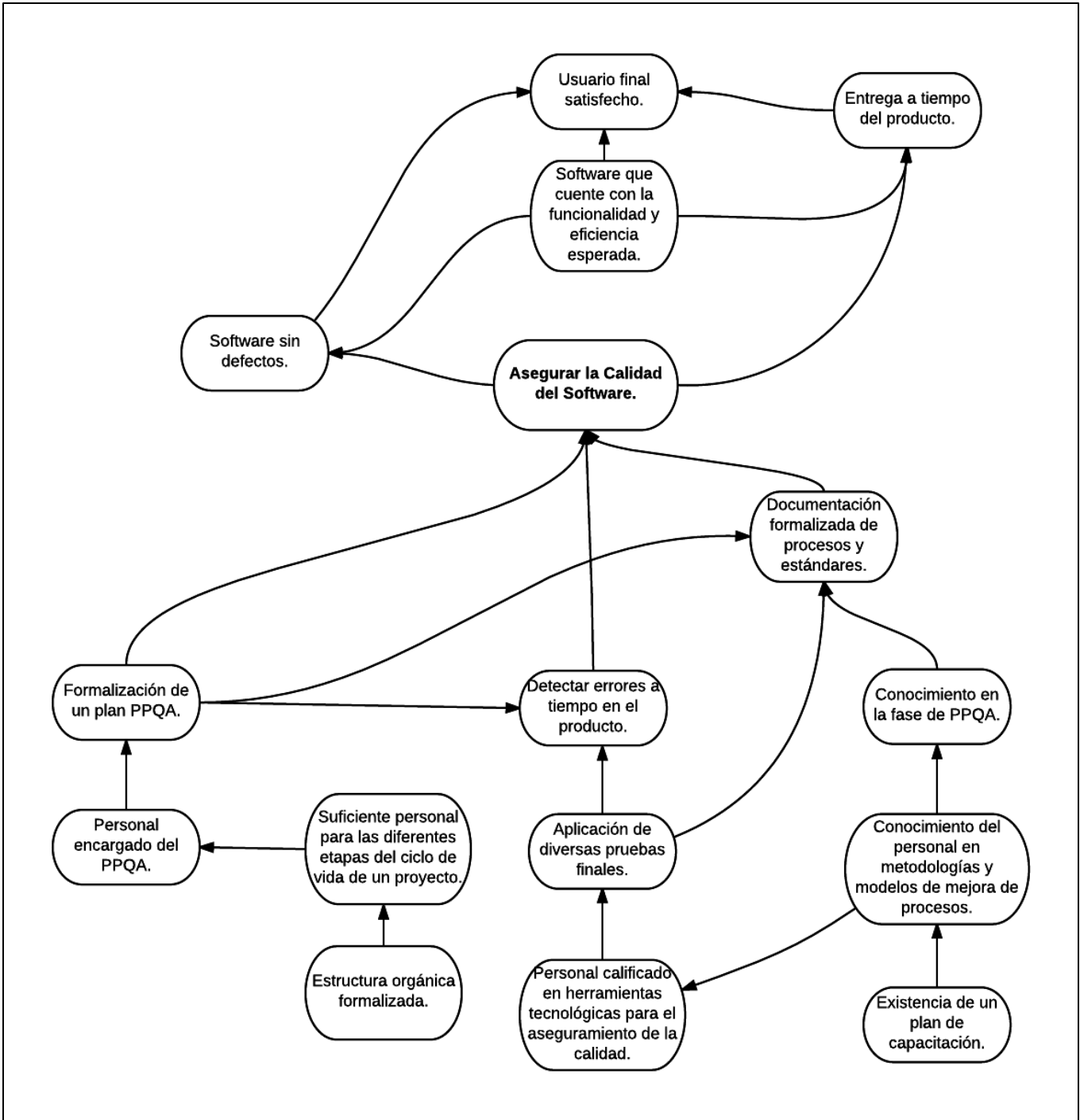


Figura 21. Árbol de Objetivos

FASE 3: ESTABLECIMIENTO

Versión	Fecha	Autor
1.4	19/12/2016	Lady Iñiguez Carchi

Objetivos

Los objetivos de esta fase son:

- Definir roles y responsabilidades necesarios para el área PPQA.
- Desarrollar el Plan de PPQA definiendo políticas, procedimientos, procesos.

Técnicas

- Análisis de material bibliográfico
- Reunión con los involucrados

Involucrados

- Miembros del GDS.
- Director de Trabajo de Titulación.
- Tesista.

Problemas y Soluciones

Problema 1: Poco conocimiento del cómo plantear los procesos de Aseguramiento de la Calidad del Proceso y del Producto (PPQA).

Solución 1: Mucha investigación y consultas a personas con conocimientos del tema.

Problema 2: Dificultades al elegir las herramientas óptimas para las pruebas especificadas

Solución 2: Mucha investigación y consultas a personas con conocimientos del tema.

Glosario de Términos

- CMMI, Capability Maturity Model Integration, Modelo de Capacidad de Madurez Integrada.
- PPQA, Aseguramiento de la Calidad del Proceso y del Producto.
- SPI, Software Process Improvement, Mejora de Procesos Software.
- BPMN, Business Process Model and Notation
- GDS, Grupo de Desarrollo Software.

- UTI, Unidad de Telecomunicaciones e Información.
- IDEAL, Iniciar, Diagnosticar, Ejecutar, Aprender,
- QA, Aseguramiento de la Calidad
- QC, Control de Calidad

6.3. FASE ESTABLECIMIENTO: Desarrollar el Plan de SQA.

6.3.1. Introducción

En esta fase se desarrolla la propuesta de mejora de procesos basado en el Modelo de Capacidad de Madurez Integrada CMMI v1.3 específicamente en el área de Aseguramiento de la Calidad del Proceso y del Producto (PPQA), en dicha propuesta se detallará un plan de PPQA, se definirán roles y responsabilidades de los miembros que conformarán el grupo de PPQA; así también como la definición de políticas, procedimientos y procesos.

6.3.2. Roles y Responsabilidades

Definir roles y responsabilidades de los distintos miembros que conforma un grupo de PPQA es de vital importancia para iniciar con el plan de SPI. Se ha tomado como base la información proporcionada por el Comité Internacional de Cualificaciones de Pruebas de Software (ISTQB) [74] [75] para proponer roles, responsabilidades y políticas de calidad, los mismos que son:

- Líder del Proyecto
- Líder de PPQA
- Tester

Líder del Proyecto

El Líder del Proyecto tiene las siguientes responsabilidades:

- Planifica las actividades del proyecto.
- Coordina conjuntamente con la alta Gerencia las iteraciones con los clientes.
- Define la organización y estructura del proyecto.
- Administra recursos necesarios para cumplir los plazos establecidos.
- Organiza el equipo de trabajo para lograr el objetivo definido.
- Estudia la viabilidad de los requisitos en unión con el Analista.
- Planifica y realiza reuniones con el equipo de trabajo en el tiempo establecido.
- Revisa junto al Líder de PPQA los avances de acuerdo al plan de PPQA.
- Informa sobre el estado actual del proyecto a los clientes y a la alta Gerencia.
- Registra formalmente el avance del proyecto.

Líder de PPQA

El Líder de PPQA tiene las siguientes responsabilidades:

- Planifica las actividades de PPQA.
- Gestiona los recursos necesarios para la ejecución del Plan de PPQA.
- Ajusta el plan de PPQA a las necesidades específicas del proyecto.
- Asegura que se cumplan los estándares durante el proceso de desarrollo.
- Realiza el seguimiento de las actividades planificadas de PPQA.
- Informa al Líder del proyecto acerca de los resultados de las actividades de PPQA.
- Reporta periódicamente al Líder de Proyecto y a la alta Gerencia sobre las actividades de PPQA.
- Valida los informes de PPQA para el paso a producción de un software.

Tester

El Tester tiene las siguientes responsabilidades:

- Ejecuta las tareas definidas en el Plan de PPQA.
- Propone casos de prueba de ser necesario.
- Ejecuta casos de prueba.
- Garantiza el cumplimiento de los requisitos, casos de uso y prototipo propuestos al inicio del proyecto.
- Analiza y notifica errores encontrados en la ejecución.
- Entrega informes de revisión al Líder de PPQA.

6.3.3. Definición de políticas para el PPQA

Las políticas de calidad deben estar enfocadas en las metas de la organización, es por ello que dichas políticas permitirán realizar un adecuado PPQA en los proyectos de desarrollo de software siendo estas:

- Establecer un plan de Proyecto para el desarrollo del proyecto.
- Mantener reuniones constantes con el Director, Subdirector y equipo de trabajo para informar el avance del proyecto.
- Establecer un plan de PPQA para un proyecto ajustándose al mismo.
- Definir un grupo SPI con roles y responsabilidades definidas para coordinar y asegurar la calidad del proyecto.
- Revisar las actividades del plan de PPQA con el Líder de Proyecto.
- El Líder de PPQA debe informar periódicamente las actividades realizadas del plan de PPQA a la Gerencia.
- Regirse al plan de PPQA para ejecutar las actividades.

- Verificar que las actividades del software se rijan a los requerimientos y procedimientos establecidos.
- Se pasará a producción un proyecto si el Líder de PPQA aprueba la funcionalidad.
- El Líder de PPQA planificará los casos de prueba dependiendo de las pruebas a realizar.
- El grupo de PPQA deberá documentar las no-conformidades encontradas durante todo el proceso de AQ.

6.3.4. Flujo de Procesos

Los procesos que a continuación se indican son los procesos que fueron entregados por el Subdirector de Desarrollo de Software Ing. Edison Coronel, mismos con los que actualmente el Grupo de Desarrollo de Software de la UTI desarrolla sus proyectos, no siguiendo a cabalidad cada una de sus fases, sino más bien tratando de acoplarse lo más posible a los mismos dicho por el mismo Subdirector de Desarrollo de Software, estos procesos hoy por hoy no están debidamente formalizados dentro de la Unidad de Telecomunicaciones e Información.

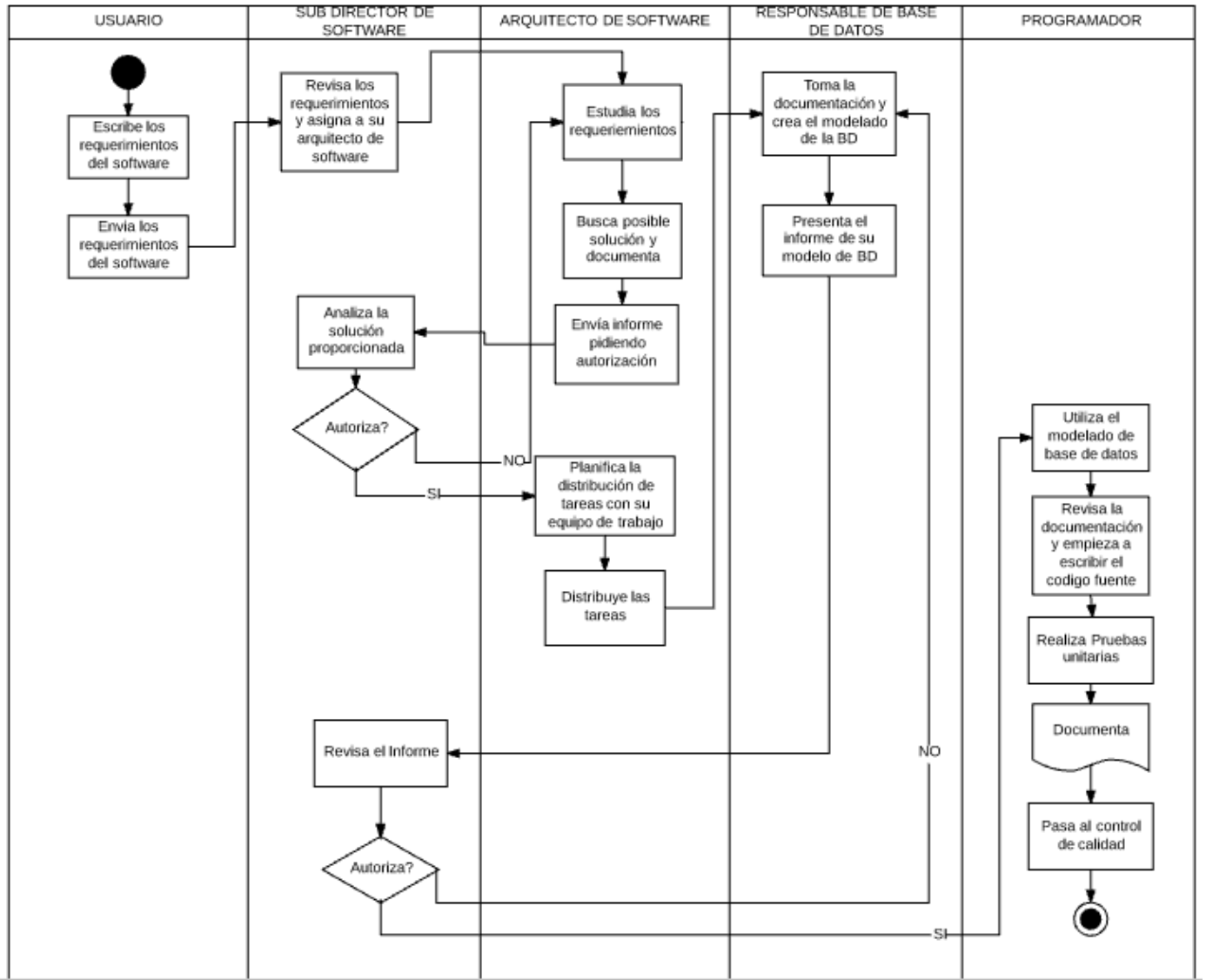


Figura 22. Proceso actual de Desarrollo de Software del GDS

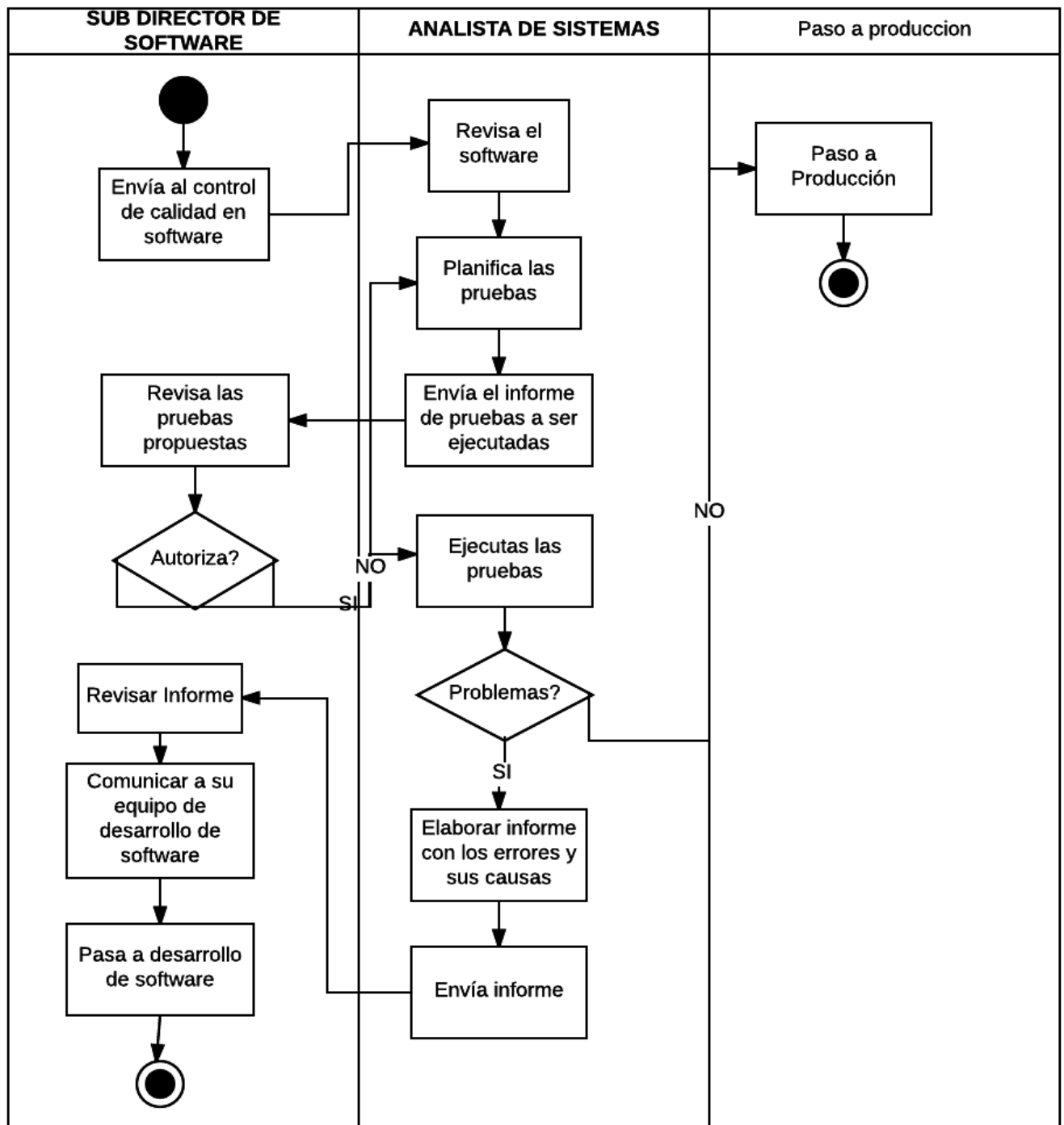


Figura 23. Proceso actual de Control de Calidad del GDS

A continuación se detallan los procesos que son la propuesta que se realiza para el GDS mejorando la versión actual de los mismos, se diseñaron siguiendo el ciclo de vida en cascada y el estándar BPMN los mismos que fueron aprobados por el Subdirector de Desarrollo de Software (**Anexo 6**), desarrollados en base a las metas y prácticas específicas que proporciona CMMI DEV-v1.3.

• Proceso Desarrollo de Software

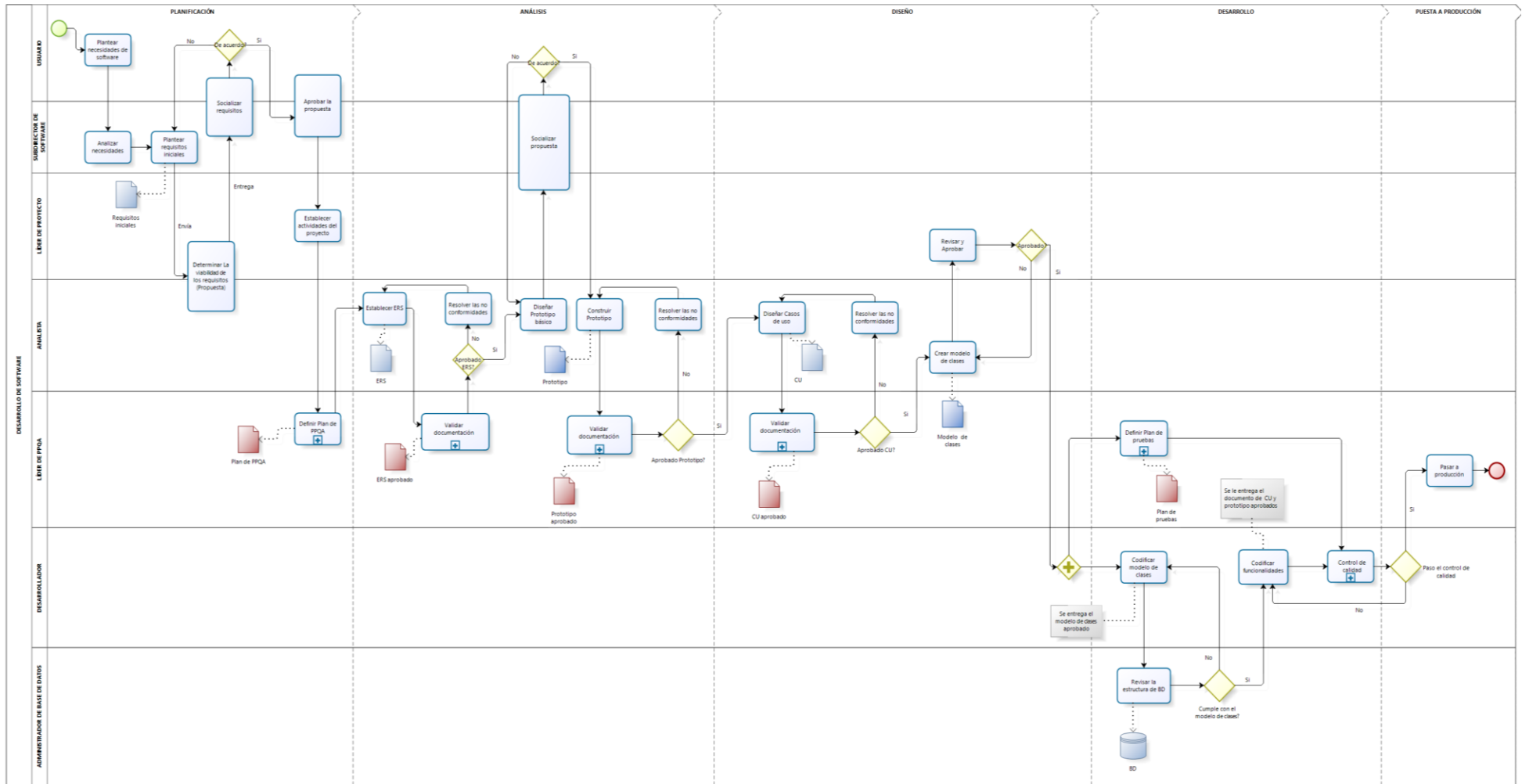


Figura 24. Proceso de Desarrollo de Software

TABLA XVIII. ESPECIFICACIÓN DEL PROCESO DE DESARROLLO DE SOFTWARE

NOMBRE DEL PROCESO	Desarrollo de Software
Versión	1.0
Código	001
Objetivo	El objetivo de este procedimiento es describir los pasos a seguir en todo el ciclo de vida del software para desarrollar un producto software de calidad.
Actores	Subdirector de software Líder de Proyecto Líder de PPQA Analista Diseñador de Base de Datos Desarrollador Usuario
Criterios de Entradas	
<ul style="list-style-type: none"> • Necesidades del cliente/usuario 	
Criterios de aprobación de los requisitos	
<ul style="list-style-type: none"> • El subdirector de Software aprobará la propuesta (requisitos) si cumple con los siguientes lineamientos [76]: <ul style="list-style-type: none"> ➤ Todo requerimiento debe ser corrector, es decir debe reflejar alguna necesidad real del sistema deseado. ➤ No existe ambigüedad en los requerimientos. ➤ Ser concisos y fáciles de entender. 	
Secuencia del Procedimiento	<ol style="list-style-type: none"> 1. El Usuario plantea sus necesidades al Subdirector de software. 2. El Subdirector de software analiza dichas necesidades y plantea una lista inicial de posibles requisitos 3. El Subdirector de software envía al Líder de proyecto quién con el Analista determinarán la viabilidad de los requisitos (propuesta). <ol style="list-style-type: none"> 3.1. El Líder de proyecto entrega al Subdirector de software la propuesta. 3.2. El Subdirector de software socializa la propuesta con el Usuario. 3.3. En caso de que el Subdirector de Software y el Usuario no estén de acuerdo, se volverá a plantear los requisitos iniciales.

	<p>3.4. Si el Usuario y el Subdirector de Software está de acuerdo el Subdirector junto con el usuario aprobarán la propuesta.</p> <p>4. El Líder de Proyecto planifica las actividades del proyecto.</p> <p>5. El Líder de PPQA define el plan de PPQA. Tabla (001)</p> <p>6. El Analista establece el ERS y lo envía a validar documentación. Tabla (002)</p> <p>6.1. En caso de no validar la documentación se informará al Analista para que resuelva las no conformidades encontradas.</p> <p>6.2. En caso de ser válido el ERS el Analista diseñará los prototipos básicos.</p> <p>7. El Analista entrega al Líder de Proyecto, para que junto con el Subdirector de Software y el Usuario socialicen el prototipo.</p> <p>7.1. En caso de no aprobar el prototipo el Usuario se informará al Analista para que modifique el diseño del prototipo.</p> <p>7.2. En caso de ser válido el Analista documentará formalmente el prototipo y lo envía a validar documentación. Tabla (002)</p> <p>7.3. En caso de no validar la documentación se informará al Analista para que resuelva las no conformidades encontradas.</p> <p>7.4. En caso de ser válido el prototipo el Analista diseñará los casos de uso.</p> <p>8. Nuevamente el Analista envía los casos de uso a validar documentación. Tabla (002)</p> <p>8.1. En caso de no validar la documentación se informará</p>
--	--

	<p>al Analista para que resuelva las no conformidades.</p> <p>8.2. En caso de ser válido los CU el Analista diseñará el modelo de clases.</p> <p>8.3. Se le envía al Líder de Proyecto para que revise y apruebe.</p> <p>8.3.1. En caso de no aprobar se modifica el modelo de clases.</p> <p>8.3.2. En caso de ser aprobado se inicia a Codificar el modelo de clases y a la vez definir el Plan de pruebas.</p> <p>Tabla (003)</p> <p>9. El Administrador de Base de Datos revisa la estructura de la BD.</p> <p>9.1. Si el Desarrollador no sigue el modelo de clases se vuelve a codificar el modelo.</p> <p>9.2. Si el Desarrollador sigue el modelo de clases se empieza a desarrollar las funcionalidades.</p> <p>10. El Desarrollador envía el software (por módulos, funcionalidad o aplicación completa) a Control Calidad para ser validado. (Tabla 004)</p> <p>10.1. Si es validado pasa a producción.</p> <p>10.2. De no ser así se enviará nuevamente a desarrollador para corregir errores.</p>
<p>Criterios de Salidas: Producto software de calidad</p>	

- **Proceso Plan de Aseguramiento de la Calidad del Proceso y del Producto (PPQA)**

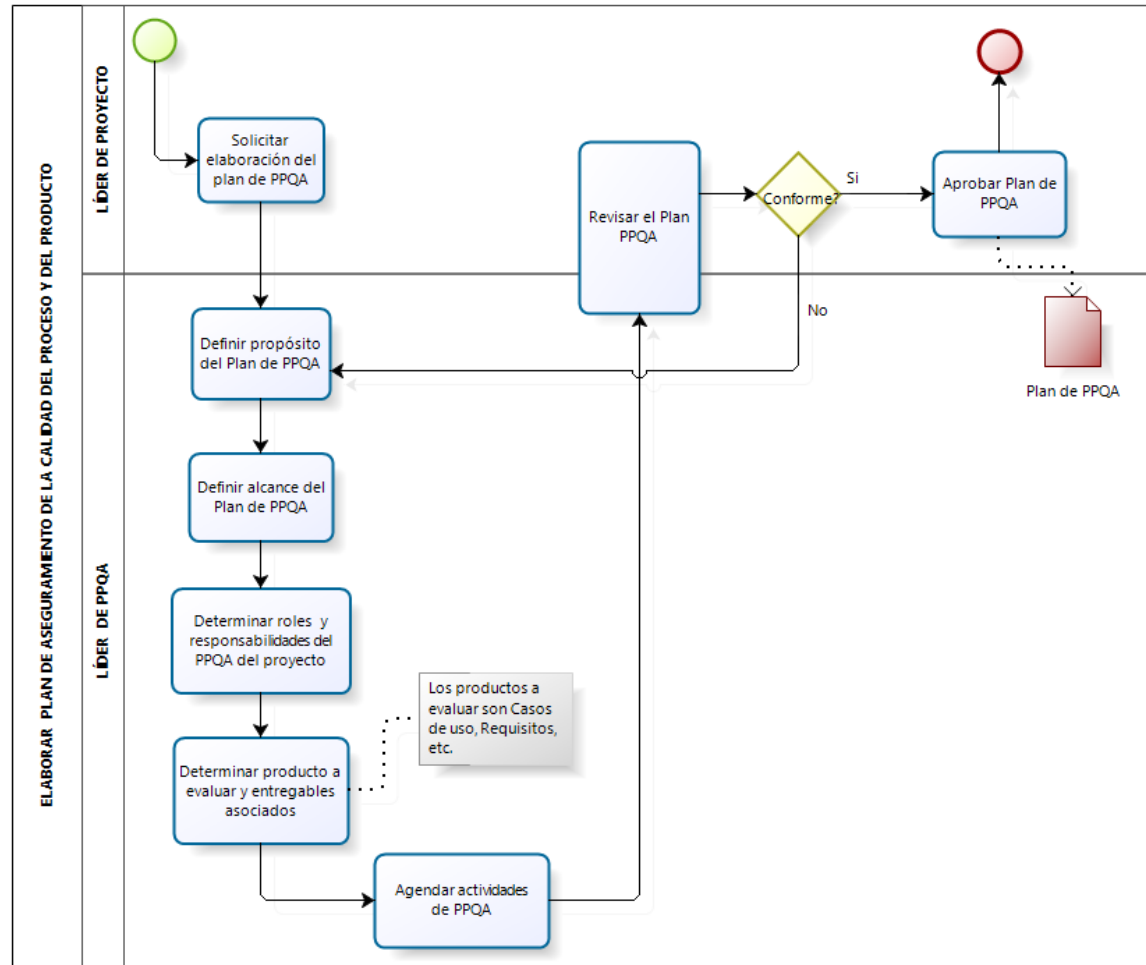


Figura 25. Proceso Plan de Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

TABLA XIX. ESPECIFICACIÓN DEL PROCESO PPQA

NOMBRE DEL PROCESO	Definición del Plan de Aseguramiento de la Calidad del Proceso y del Producto (PPQA).
Versión	1.0
Código	001
Objetivo	El objetivo de este procedimiento es describir los pasos a seguir para definir las actividades dentro del plan de PPQA.
Actores	Líder de Proyecto Líder de PPQA
Criterios de Entradas <ul style="list-style-type: none"> • Ninguno 	
Secuencia del Procedimiento	<ol style="list-style-type: none"> 1. El Líder de Proyecto solicita la elaboración del plan de PPQA al Líder de PPQA. 2. El Líder de PPQA define el propósito del plan de PPQA. 3. Define también el alcance del plan de PPQA. 4. Se determina roles y responsabilidades del PPQA del proyecto. 5. Se determina productos a evaluar y entregables asociados. 6. Por último el Líder de PPQA agenda actividades de PPQA. 7. El Líder de PPQA junto con el Líder de Proyecto revisan el plan de PPQA. <ol style="list-style-type: none"> 7.1. En caso de no aprobar el Líder de Proyecto el plan de PPQA el Líder de PPQA deberá diseñar un nuevo plan. 7.2. En caso de estar todo correcto el Líder de Proyecto aprobará el plan de PPQA.
Criterios de Salidas: Plan de PPQA	

- Proceso Validar Documentación

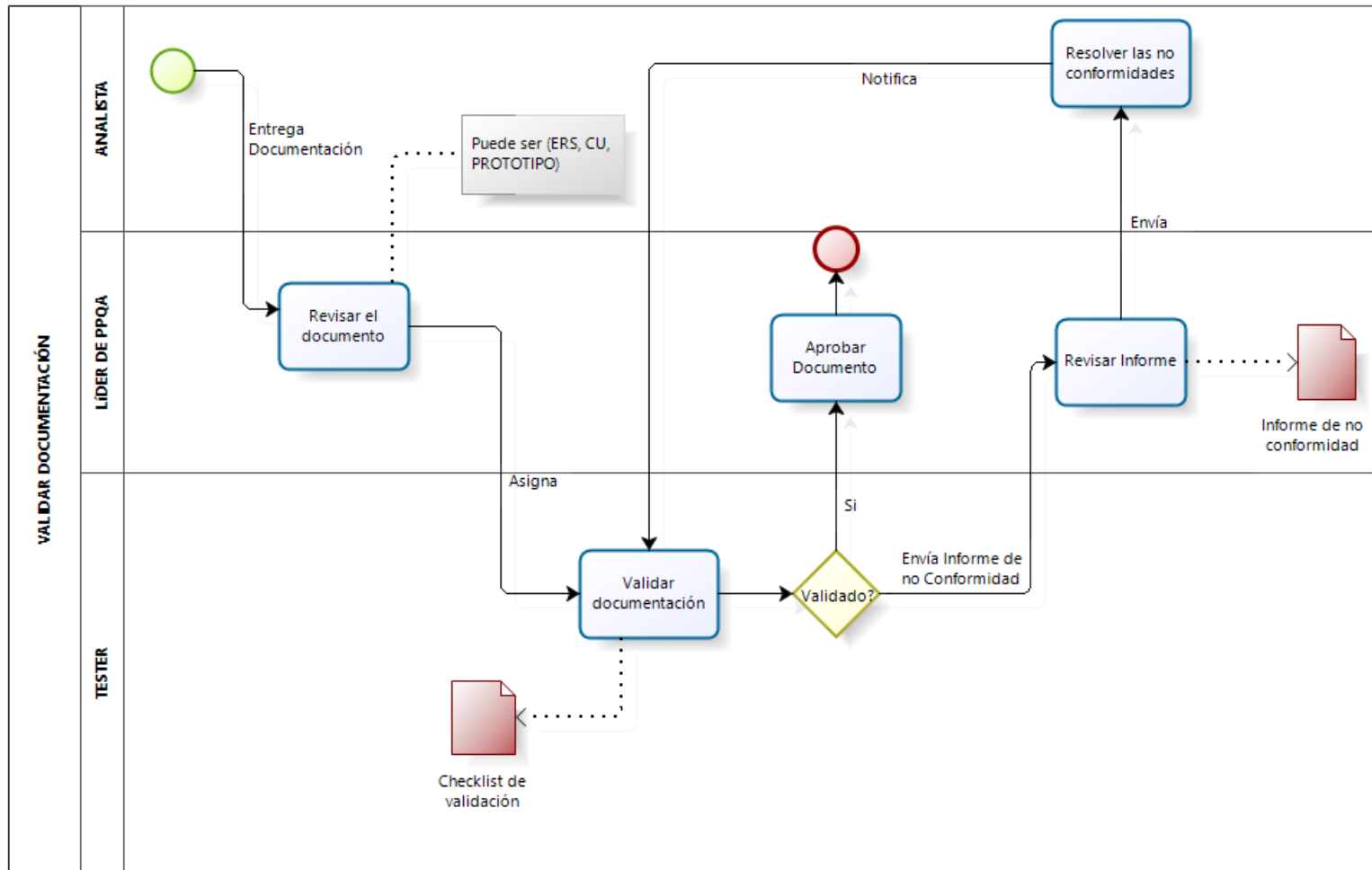


Figura 26. Proceso Validar Documentación

TABLA XX. ESPECIFICACIÓN DEL PROCESO VALIDAR DOCUMENTACIÓN

NOMBRE DEL PROCESO	Validar Documentación
Versión	1.0
Código	002
Objetivo	El objetivo de este procedimiento es describir los pasos a seguir para validar la documentación inicial, comprobando que se refleje los puntos acordados con el cliente, documentando las no conformidades.
Actores	Líder de PPQA Analista Tester
Criterios de Entradas <ul style="list-style-type: none"> • Documento ERS • Casos de uso y Prototipo 	
Secuencia del Procedimiento	<ol style="list-style-type: none"> 1. El Analista entrega la documentación (puede ser ERS, CU y Prototipo) al Líder de PPQA. 2. El Líder de PPQA revisa la documentación y asigna un Tester que valide mediante el Checklist de validación. <ol style="list-style-type: none"> 2.1. Si la documentación no cumple con los criterios establecidos en el Checklist se notifica al Líder de PPQA mediante un informe de no conformidades. <ol style="list-style-type: none"> 2.1.1. El Líder de PPQA revisa el informe y entrega al Analista para resolver las no conformidades y volver a validar. 2.2. Si la documentación cumple con los criterios establecidos en el Checklist el Líder de PPQA aprueba la documentación.
Criterios de Salidas: Documentación validada	

- Proceso Plan de Pruebas

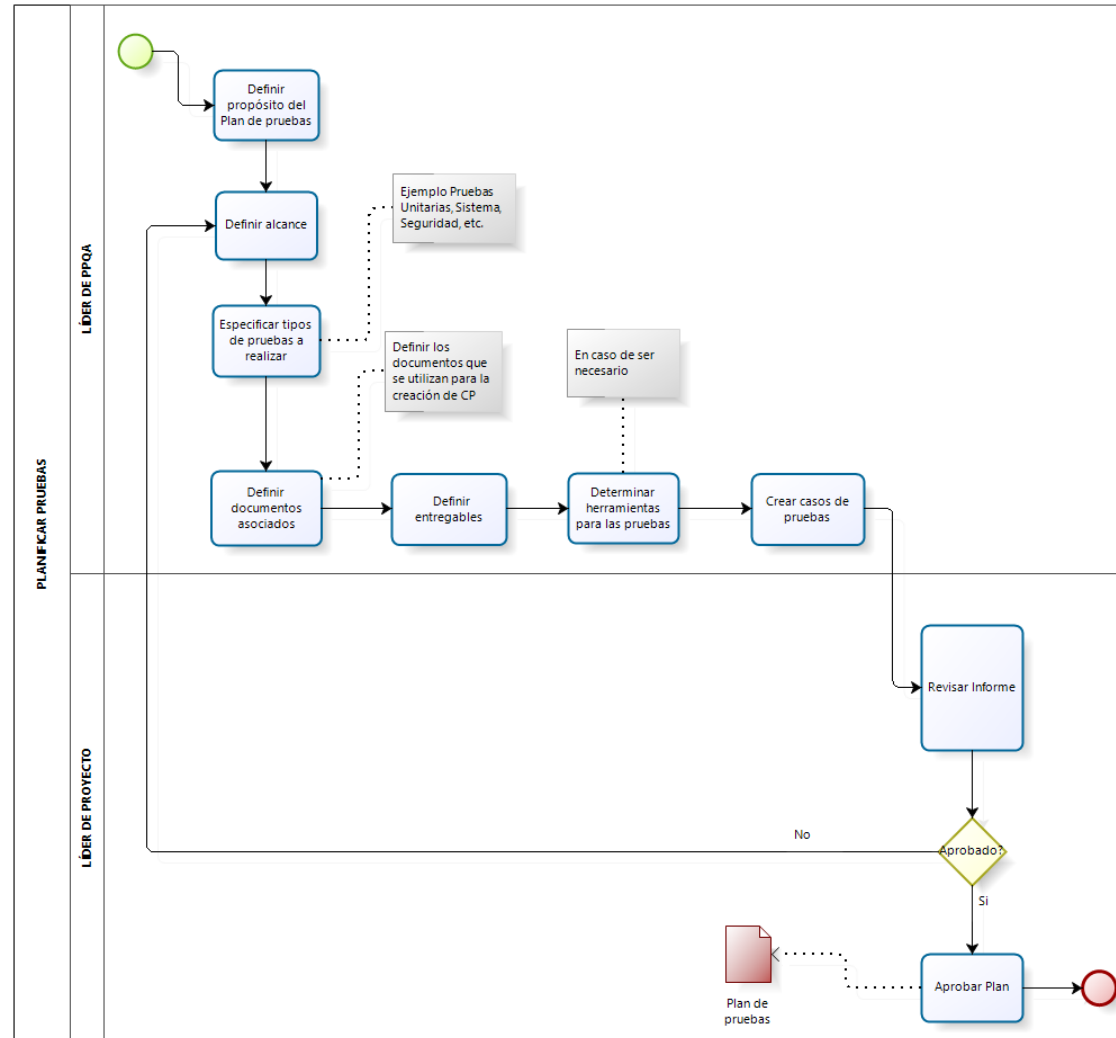


Figura 27. Proceso Plan de Pruebas

TABLA XXI. ESPECIFICACIÓN DEL PROCESO PLAN DE PRUEBAS

NOMBRE DEL PROCESO	Plan de Pruebas.
Versión	1.0
Código	003
Objetivo	El objetivo de este procedimiento es describir los pasos a seguir para diseñar el plan de pruebas.
Actores	Líder de Proyecto Líder de PPQA
Criterios de Entradas <ul style="list-style-type: none"> • Ninguno 	
Secuencia del Procedimiento	<ol style="list-style-type: none"> 1. El Líder de PPQA define el propósito del plan de pruebas. 2. Así también define el alcance del mismo. 3. Se especifica los tipos de pruebas a realizar. 4. Se define los documentos asociados, los mismos que me permitirán crear los casos de prueba. 5. También se define los entregables que se obtendrá con la realización de las pruebas. 6. Determinar herramientas para las pruebas, solo en caso de ser necesario. 7. Por último el Líder de PPQA crea los casos de prueba dependiendo de los tipos de prueba a realizar. 8. El Líder de PPQA revisa el informe completo y aprueba. <ol style="list-style-type: none"> 8.1 Si se aprueba se obtendrá el Plan de pruebas. 8.2. En caso de no aprobarse el Líder de PPQA volverá a definir el plan de pruebas.
Criterios de Salidas: Plan de pruebas	

- Proceso Control de Calidad

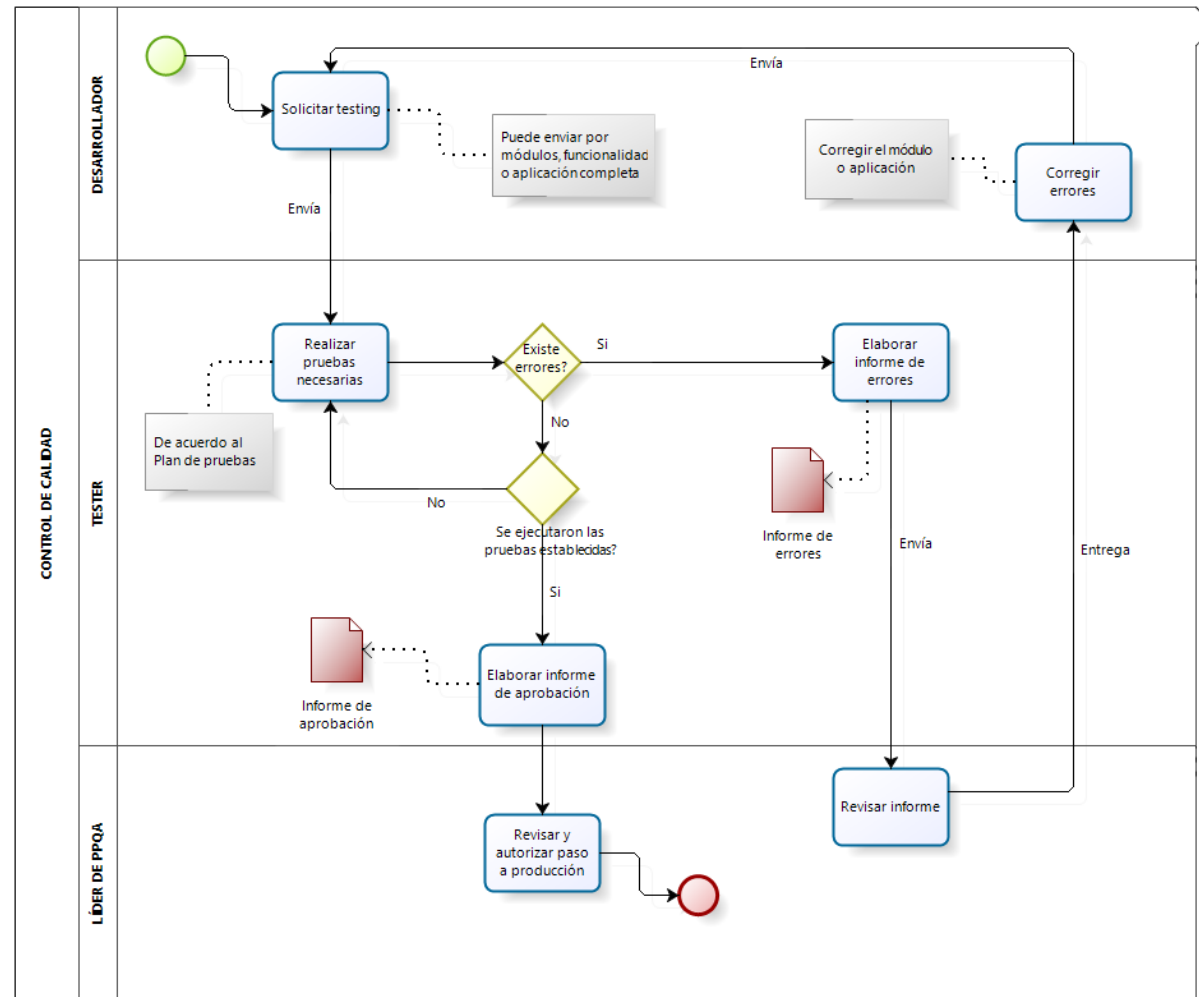


Figura 28. Proceso Control de Calidad

TABLA XXII. ESPECIFICACIÓN DEL PROCESO CONTROL DE CALIDAD

NOMBRE DEL PROCESO	Control de calidad.
Versión	1.0
Código	004
Objetivo	El objetivo de este procedimiento es describir los pasos a seguir para el control de calidad del producto mediante la ejecución de casos de prueba.
Actores	Desarrollador Tester Líder de PPQA
Criterios de Entradas <ul style="list-style-type: none"> • Plan de pruebas 	
Secuencia del Procedimiento	<ol style="list-style-type: none"> 1. El desarrollador solicita que se realice el testing ya sea de un módulo, funcionalidad o aplicación completa. 2. El Tester procede a realizar las pruebas necesarias, las mismas que ya se plasmaron en el plan de pruebas. 3. Si existen errores en cualquiera de las pruebas antes establecida el Tester elabora un informe de errores. <ol style="list-style-type: none"> 3.1. Se envía al Líder de PPQA el revisa dicho informe y envía al Desarrollador para que corrija y pueda solicitar nuevamente el testing. 3.2. En caso de no existir errores el Tester elabora un informe de pruebas exitosas y envía al Líder de PPQA. 3.3. El Líder de PPQA revisa dicho informe y autoriza el paso a producción.
Criterios de Salidas: Producto software con calidad	

6.3.5. Productos de trabajo – Plantillas

El levantamiento de plantillas se hace de acuerdo a los procesos antes planteados en los cuales se visualiza productos de trabajo de un color marrón, los mismos que son las plantillas que se va a proponer.

Para determinar cuáles fueron los aspectos más importantes a tomar en cuenta para el diseño de las plantillas se realizó una búsqueda extensa, puesto que CMMI como IDEAL no brindan un soporte para el planteamiento de las mismas valiéndose de la investigación tanto de trabajos desarrollados como de documentación expuesta en internet, proponiendo las siguientes plantillas, las mismas que fueron aprobadas por la Analista/Desarrolladora Ing. Lissette López. **(Anexo 7)**

Para cada una de las plantillas se hizo uso de estándares los mismos que se acoplaron a las necesidades requeridas en el presente proyecto y sobre todo tratando de guiarnos o regirnos a la manera de trabajo de la UTI. Los estándares utilizados son:

- Plantilla plan de PPQA, estándar IEEE 730-2002.
- Plantilla Checklist de Validación de requisitos, estándar IEEE 830-1998.
- Plantilla Checklist de Validación de casos de uso, estándar UML.
- Plantilla plan de pruebas, estándar ISO/IEEE 29119-3:2013.
- Plantilla informe de errores, estándar ISO/IEEE 29119-3:2013

PLANTILLA DEL PLAN DE ASEGURAMIENTO DE LA CALIDAD DEL PROCESO Y DEL PRODUCTO



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Plan de Aseguramiento de la Calidad del Proceso y del Producto
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
APROBADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. PROPÓSITO
2. ALCANCE
3. ROLES Y RESPONSABILIDADES
4. PRODUCTOS A EVALUAR Y ENTREGABLES
5. PLANIFICAR ACTIVIDADES DE QA

1. PROPÓSITO

[En esta sección se indicará el propósito de la elaboración de este informe.]

Las características planteadas en este documento, se basan en el estándar IEEE 730-2002 [76] que describe los aspectos que debe tener un plan de Aseguramiento de la Calidad (QA), el mismo que se sirve como guía para poder plantear el plan de Aseguramiento de la Calidad del Proceso y del Producto (PPQA) de acuerdo a las especificaciones del mismo.

2. ALCANCE

[En esta sección se indicará el alcance que deberá tener el desarrollo de este informe.]

3. ROLES Y RESPONSABILIDADES

[Se definirá roles y responsabilidades del personal encargado del aseguramiento de la calidad del proyecto.]

Rol	Nombre	Responsabilidad
<cargo>	<Nombre>	

4. PRODUCTOS A EVALUAR Y ENTREGABLES

[En esta sección se determinarán los productos a evaluar y los entregables asociados.]

Producto	Rol	Responsable	Entregable asociado
	<cargo>	<nombre>	

5. PLANIFICAR ACTIVIDADES DE QA

[En esta sección se determinará las actividades de QA señalando fechas tentativas.]

Actividad	Descripción	Fecha inicio	Fecha fin

Elaborado

[Nombre y Apellido]

Aprobado

[Nombre y Apellido]

PLANTILLA CHECKLIST DE VALIDACIÓN DE REQUISITOS



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Checklist de Validación de Requisitos
VERSIÓN	1.0

VALIDADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
APROBADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. CHECKLIST DE REVISIÓN
4. CONCLUSIÓN

1. DESCRIPCIÓN

[En esta sección se redactará brevemente el contenido de este informe.]

Los criterios de revisión planteados en el checklist se basan en el estándar IEEE 830-1998 [77], el mismo que no obliga a seguir estrictamente el formato pero si se debe tratar de incluir de una forma o de otra, toda la información posible.

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. CHECKLIST DE REVISIÓN

[En esta sección se validarán los diversos criterios establecidos.]

CRITERIOS DE REVISIÓN	REQ1		REQ...		REQn	
	SI	NO	SI	NO	SI	NO
Los requisitos tienen un código de identificación						
Los requisitos tienen un nombre						
Los requisitos tiene una descripción						
Los requisitos están clasificados como requisitos Funcionales y no Funcionales						
Los requisitos no son ambiguos						
Los requisitos son correctos (reflejar una necesidad real)						
Los requisitos son completos (proporcionan la información suficiente para su comprensión)						
Los requisitos son concisos y fáciles de entender						
Los requisitos están clasificados por prioridad (Alta o Media)						

4. CONCLUSIÓN

[En esta sección se indicará el estado de la documentación luego de realizada la validación.]

Validado
[Nombre y Apellido]

Aprobado
[Nombre y Apellido]

PLANTILLA INFORME DE NO CONFORMIDADES DE VALIDACIÓN DE REQUISITOS



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	No-Conformidades de Validación de Requisitos
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. DESCRIPCIÓN DE NO CONFORMIDADES
4. RECOMENDACIONES

1. DESCRIPCIÓN

[En esta sección se redactará brevemente el contenido de este informe.]

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. DESCRIPCIÓN DE NO CONFORMIDADES

[En esta sección se detallan las no conformidades encontradas en la validación del documento de requisitos.]

4. RECOMENDACIONES

[En esta sección se escribirán sugerencias para mejorar el planteamiento de requisitos]

Elaborado

[Nombre y Apellido]

Revisado

[Nombre y Apellido]

PLANTILLA CHECKLIST DE VALIDACIÓN DE PROTOTIPO



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Checklist de Validación de Prototipo
VERSIÓN	1.0

VALIDADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
APROBADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. CHECKLIST DE REVISIÓN
4. CONCLUSIÓN

1. DESCRIPCIÓN

[En esta sección se indicara el propósito de la elaboración de este informe.]

Los criterios de revisión planteados en el checklist son en base a investigaciones realizadas [78] [79] de los distintos planteamientos para diseñar prototipos de pantallas, extrayendo lo más relativo y común debido a que depende mucho de la experiencia del diseñador a la hora de realizar el prototipo. Esta información sirvió como hincapié para plantear tales criterios, guiándonos a los requerimientos del proyecto en ejecución.

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. CHECKLIST DE REVISIÓN

[En esta sección se validara los diversos criterios establecidos.]

CRITERIOS DE REVISIÓN	PROT1		PROT...		PROTn	
	SI	NO	SI	NO	SI	NO
El prototipo tienen un código de identificación						
La pantalla tienen un título que la identifique						
En caso de existir los mensajes de información presentados están situados en una parte de la pantalla donde no interfieren con otros elementos.						
El tamaño de la fuente en toda la pantalla es legible a distancia						
El color de la fuente da contraste con el fondo de la pantalla.						
Todos los componentes que conforman la pantalla están alineados de forma organizada (botones, input, etiquetas)						

La ortografía de los siguientes elementos es correcta						
Botones						
Etiquetas (labels, mensajes informativos...)						
Input						

4. CONCLUSIÓN

[En esta sección se indicara el estado de la documentación luego de realizada la validación.]

Validado

[Nombre y Apellido]

Aprobado

[Nombre y Apellido]

PLANTILLA INFORME DE NO CONFORMIDADES DE VALIDACIÓN DE PROTOTIPO



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	No-Conformidades de Validación de Prototipo
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. DESCRIPCIÓN DE NO CONFORMIDADES
4. RECOMENDACIONES

1. DESCRIPCIÓN

[En esta sección se redactará brevemente el contenido de este informe.]

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. DESCRIPCIÓN DE NO CONFORMIDADES

[En esta sección se detallan las no conformidades encontradas en la validación del documento de prototipo.]

4. RECOMENDACIONES

[En esta sección se indicará el estado de la documentación luego de realizada la validación.]

Elaborado

[Nombre y Apellido]

Revisado

[Nombre y Apellido]

PLANTILLA CHECKLIST DE VALIDACIÓN DE CASOS DE USO



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Checklist de Validación de Casos de Uso
VERSIÓN	1.0

VALIDADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
APROBADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. CHECKLIST DE REVISIÓN
4. CONCLUSIÓN

1. DESCRIPCIÓN

[En esta sección se indicará el propósito de la elaboración de este informe.]

Los criterios de revisión planteados en el checklist se basan en el estándar UML [80], el mismo que permite especificar, visualizar, construir y documentar artefactos de un sistema de software.

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. CHECKLIST DE REVISIÓN

[En esta sección se validará los diversos criterios establecidos.]

CRITERIOS DE REVISIÓN	CU1		CU...		CUn	
	SI	NO	SI	NO	SI	NO
Los casos de uso tiene un código de identificación						
Los casos de uso tienen un nombre						
El nombre refleja algo que el sistema debe hacer						
Los casos de uso tienen una breve descripción						
La descripción es clara						
Los casos de uso hacen referencia a los requisitos asociados						
Los casos de uso hacen referencia al prototipo asociado						
Los casos de uso tienen una precondición						
Los casos de uso tienen un postcondición						
Los casos de uso tienen un flujo principal de eventos						
Se contempla todos los posibles flujos alternos						
Los pasos de los flujos tiene una secuencia lógica						

4. CONCLUSIÓN

[En esta sección se indicará el estado de la documentación luego de realizada la validación.]

Validado

[Nombre y Apellido]

Aprobado

[Nombre y Apellido]

PLANTILLA INFORME DE NO CONFORMIDADES DE VALIDACIÓN DE CASOS DE USO



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	No-Conformidades de Validación de Casos de Uso
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. DESCRIPCIÓN DE NO CONFORMIDADES
4. RECOMENDACIONES

1. DESCRIPCIÓN

[En esta sección se redactará brevemente el contenido de este informe.]

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. DESCRIPCIÓN DE NO CONFORMIDADES

[En esta sección se detallan las no conformidades encontradas en la validación del documento de casos de uso.]

4. RECOMENDACIONES

[En esta sección se escribirán sugerencias para mejorar el planteamiento de casos de uso]

Elaborado

[Nombre y Apellido]

Revisado

[Nombre y Apellido]

PLANTILLA PLAN DE PRUEBAS



UNIDAD DE TELECOMUNICACIONES E INFORMACIÒN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Plan de Pruebas
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
APROBADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. PROPÓSITO
2. ALCANCE
3. DOCUMENTOS ASOCIADOS
4. ENTREGABLES
5. TIPOS DE PRUEBAS A REALIZAR
6. HERRAMIENTAS A UTILIZAR
7. CASOS DE PRUEBAS
8. GLOSARIO

1. PROPÓSITO

[En esta sección se indicará el propósito de la elaboración de este informe.]

Las características planteadas en este documento, se establecen en el estándar ISO/IEC/IEEE 29119-3:2013 [81] que describe los aspectos que deben tener un plan de pruebas, informe de estado de pruebas, informe de terminación de pruebas, basándose específicamente en el plan de pruebas (**Anexo 8**) el mismo que se acoplo a las necesidades del proyecto en ejecución.

2. ALCANCE

[En esta sección se indicará el alcance que deberá tener el desarrollo de este informe.]

3. TIPOS DE PRUEBAS A REALIZAR

[En esta sección se definirán las pruebas óptimas a realizar.]

4. DOCUMENTOS ASOCIADOS

[En esta sección se definirán los documentos que se utilizará para diseñar los casos de pruebas.]

5. ENTREGABLES

[En esta sección se definirá los documentos a entregar en la ejecución de los casos de pruebas.]

6. HERRAMIENTAS A UTILIZAR

[En esta sección se detallará la o las herramientas a ser utilizadas en caso de ser necesario en la ejecución de los casos de pruebas.]

7. CASOS DE PRUEBA

[En esta sección se detallará los casos de pruebas a ejecutar.]

8. GLOSARIO

[En esta sección se definirá la terminología no común utilizada en el documento.]

Elaborado

Lady Iñiguez

Aprobado

Lissette López

PLANTILLA INFORME DE ERRORES



UNIDAD DE TELECOMUNICACIONES E INFORMACIÒN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Informe de errores
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. DESCRIPCIÓN
2. OBJETIVOS
3. REPORTE DE ERRORES

1. DESCRIPCIÓN

[En esta sección se redactará brevemente el contenido de este informe.]

Las características planteadas en este documento, se establecen en el estándar ISO/IEC/IEEE 29119-3:2013 [81] que describe los aspectos que debe tener un informe de errores el mismo que se acoplo a las necesidades del proyecto en ejecución.

2. OBJETIVOS

[En esta sección se especificará los objetivos a alcanzar con la elaboración de este informe.]

3. REPORTE DE ERRORES

Reporte de prueba Nro: _____			
Fecha de prueba: dd/mm/aaaa		Tipo de prueba: _____	
Id. Caso de prueba	Pasos	Nivel de Severidad	Descripción

Elaborado

[Nombre y Apellido]

Revisado

[Nombre y Apellido]

PLANTILLA INFORME DE APROBACIÓN



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	<Nombre del Proyecto>
LÍDER DE PROYECTO	<Nombre>
FECHA DE INICIALIZACIÓN	<dd/mm/aaaa>

NOMBRE DE DOCUMENTO	Informe de aprobación
VERSIÓN	1.0

ELABORADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>
REVISADO POR:	<Nombre>	
	<Cargo>	
	FECHA:	<dd/mm/aaaa>

Contenido

1. PROPÓSITO
2. REPORTE DE APROBACIÓN

1. PROPÓSITO

[En esta sección se redactará brevemente el contenido de este informe.]

2. REPORTE DE APROBACIÓN

Reporte de Aprobación
Fecha de aprobación: dd/mm/aaaa
Descripción:

Elaborado
[Nombre y Apellido]

Revisado
[Nombre y Apellido]

6.3.6. Comparativa de Herramientas

En el proceso para seleccionar una herramienta para pruebas es necesario considerar factores que en el ciclo de vida del proyecto entran a jugar como cuantas y que tipos de pruebas se va a realizar, igualmente para todo proyecto de software es necesario determinar el alcance de las pruebas, es decir, si es para un proyecto o para todos los proyectos en la organización, qué tan compleja es la aplicación a probar, o si existe una sola aplicación bajo prueba o son varias [82]. Con base a esos requisitos se determinó que las pruebas a realizar son: prueba funcional, de rendimiento y de seguridad a nivel de aplicación (**Anexo 10**). A continuación se hace un análisis comparativo de herramientas para pruebas de rendimiento, funcionales y de seguridad para determinar las más idóneas.

Existe una gran variedad de herramientas Open Source como Comerciales que realizan desde la gestión de pruebas hasta para cada tipo de prueba funcional y no funcional, por ello se pretende hacer una clasificación de las posibles herramientas a utilizar y para recomendar al GDS de la Unidad de Telecomunicaciones e Información (UTI) que se rijan a los lineamientos planteados:

- a. Más conocidas o utilizadas.
- b. Plataforma de funcionamiento.
- c. Manual de instalación.
- d. Manual de usuario.
- e. Generador de informe de pruebas.
- f. Idiomas soportados.
- g. Soporte técnico.
- h. Tipo de pruebas a realizar: pruebas funcionales, seguridad y rendimiento. (**Anexo 9**)

• Herramientas para pruebas funcionales

El objetivo de las herramientas para prueba de Funcionalidad es validar el comportamiento de la aplicación, determinando el cumplimiento de los requisitos inicialmente planteados. En base a algunos trabajos interesantes de comparación de herramientas Comerciales como Open Source [83] [84] [85], se da inicio a la siguiente clasificación haciendo una abstracción de las herramientas más utilizadas las mismas que se detallan a continuación en la TABLA XVIII:

TABLA XXIII. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS FUNCIONALES

HERRAMIENTA	Testlink	Selenium	Test Complete	Rational Functional Tester	Microsoft Test Manager	Unified Functional Tester	WET	SoapUI
Dirección url del fabricante	http://testlink.org/	http://www.seleniumhq.org/	https://smartbear.com/product/testcomplete/overview/	http://www-03.ibm.com/software/products/es/functional	https://msdn.microsoft.com/en-us/library/jj635157.aspx	http://www8.hp.com/us/en/software-solutions/unified-functional-automated-testing/	http://wet.qantom.org/home.html	https://www.soapui.org/
Comercial / Open Source	Open Source	Open Source	Comercial	Open Source	Comercial	Comercial	Open Source	Open Source
Plataforma de funcionamiento	Windows y Linux	Windows, MacOS y Linux	Windows	Windows 2000, Windows XP, y Linux	Windows	Windows	Windows	Windows, MacOS y Linux
Formato de resultados de pruebas	Office, OpenOffice, Excel, HTML Email	No se evidencia	MHT HTML, XML, mensajes, imágenes, archivos PDF	CSV, Word, Excel, HTML, XML	No se evidencia	HTML, XML	HTML	PDF, HTML, RTF, Excel, XML
Idiomas soportados	Ingles Español	Ingles	Ingles	Ingles Español Italiano Francés Coreano Japonés Portugués	Ingles Español	Ingles	Ingles	Ingles
Pruebas Manuales/Automáticas	Manuales	Automáticas	Manuales/Automáticas	Automáticas	Manuales/Automáticas	Automáticas	Automáticas	Automáticas

Lenguajes soportados para la creación de scripts	No se evidencia	Java, C#, Ruby, Python	JavaScript, Python, VBScript, JScript, DelphiScript, C++, Script, y C#	.NET, Java	Visual, C#, .NET.	.NET, Java	No se evidencia	Java, C++,
Manual de instalación	http://testlink.sourceforge.net/docs/docs/toc.php?page=11	http://docs.seleniumhq.org/docs/02_selenium_ide.jsp#installing-the-ide	https://support.smartbear.com/testcomplete/docs/general-info/installing/index.html	ftp://public.dhe.ibm.com/software/rational/dp/v7/rt/7012/docs/install_instruction/pdf/RF_install_es.pdf	No se evidencia	http://www.tutorialspoint.com/qtp/qtp_tutorial.pdf	http://wet.qantom.org/InstallationGuide.html	https://www.soapui.org/getting-started/installing-soapui.html
Manual de usuario	http://www.queaustral.com.ar/descargas/MTestLink.pdf	http://www.w3ii.com/es/selenium/selenium_user_interaction.shtml	https://support.smartbear.com/testcomplete/docs/tutorials/index.html	No se evidencia	https://msdn.microsoft.com/es-es/library/jj635157(v=vs.110).aspx	http://technodivine.com/downloads/Help-and-Guides/02-QTP/05-UFT-Documentation-12.02/04-UFT_GUI_Tutorial_Web.pdf	http://wet.qantom.org/userguide/overview.html	http://www.ejje.es/contenidos/informacion/herramientas_ejie/eu_0213/adjuntos/SsoapUI.%20Manual%20de%20usuario%20v2.1.pdf
Soporte	Foros, Twitter, comunidad, videos	Grupo de usuario, sala de chat	videos, foros, ayuda en línea, solucionador de problemas	Comunidad, foros	Videos, foros, redes sociales, comunidad	Foros, ayuda en línea, solucionador de problemas, comunidad	Comunidad, wikis,	Foros, blog, videos

- **Valoración numérica de herramientas Funcionales**

TABLA XXIV. SISTEMA OPERATIVO

	Windows	Linux	MacOS	TOTAL
Testlink	✓	✓	X	2
Selenium	✓	✓	✓	3
Test Complete	✓	X	X	1
Rational Functional Tester	✓	✓	X	2
Microsoft Test Manager	✓	X	X	1
Unified Functional Tester	✓	X	X	1
WET	✓	X	X	1
SoapUI	✓	✓	✓	3

TABLA XXV. SOPORTE Y DOCUMENTACIÓN

	Manual de Instalación	Manual de usuario	Foros	Comunidad	TOTAL
Testlink	✓	✓	✓	✓	4
Selenium	✓	✓	X	X	2
Test Complete	✓	✓	✓	X	3
Rational Functional Tester	✓	✓	✓	✓	4
Microsoft Test Manager	X	✓	✓	✓	3
Unified Functional Tester	✓	✓	✓	✓	4
WET	✓	✓	X	✓	3
SoapUI	✓	✓	✓	X	3

TABLA XXVI. FORMATO DE RESULTADOS

	HTML	Excel	Email	Word	PDF	Imágenes	TOTAL
Testlink	✓	✓	✓	X	✓	X	4
Selenium	X	X	X	X	X	✓	1
Test Complete	✓	X	✓	X	✓	✓	4
Rational Functional Tester	✓	✓	X	✓	X	X	3
Microsoft Test Manager	X	X	X	X	X	✓	1
Unified Functional Tester	✓	X	X	X	X	X	1
WET	✓	X	X	X	X	X	1
SoapUI	✓	✓	X	X	✓	X	3

TABLA XXVII. REQUISITOS MINIMOS DEL MÓDULO A PROBAR

	Técnica (caja negra)	Gestión del ciclo de pruebas	Open Source	TOTAL
Testlink	✓	✓	✓	3
Selenium	X	X	✓	1
Test Complete	✓	✓	X	2
Rational Functional Tester	X	X	✓	1
Microsoft Test Manager	✓	✓	X	2
Unified Functional Tester	X	X	X	0
WET	X	X	✓	1
SoapUI	X	X	✓	1

TABLA XXVIII. RESULTADOS

	Sistema Operativo	Soporte y Documentación	Soporte de lenguajes (formato de resultados)	Requisitos mínimos	TOTAL
Testlink	2	4	4	3	13
Selenium	3	2	1	1	7
Rational Functional Tester	2	4	3	1	10
WET	1	3	1	1	6
SoapUI	3	3	3	1	10
Test Complete	1	3	4	2	10
Microsoft Test Manager	1	3	1	2	7
Unified Functional Tester	1	4	1	0	6

De las herramientas evaluadas se puede evidenciar que **Testlink y Test Complete** son las más completas en los aspectos requeridos.

Test Complete es comercial contando con una prueba gratis de 30 días y en caso que se la quiera adquirir tiene un costo aproximado de \$ 4,200.00¹, es una de las herramientas comerciales más completas y usadas no solo para pruebas funcionales sino para gestionar todo el ciclo de pruebas acoplándose a pruebas manuales o automáticas; pero rigiéndose a los lineamientos de la Unidad de Telecomunicaciones e Información (UTI) los mismos que trabajan bajo software libre.

TestLink es la herramienta seleccionada puesto que se acopla a la técnica de caja negra antes especificada (Anexo 9), además permite gestionar las pruebas desde la creación del Plan de pruebas hasta el informe de realización de pruebas siendo una herramienta completa en cuanto a la gestión de casos de prueba funcional y contando con la mayoría de soporte para su entendimiento.

¹ Test Complete página principal <https://smartbear.com/product/testcomplete/overview/> para cotizar precios

- **Herramientas para pruebas de seguridad**

Existe una gran gama de herramientas para pruebas de seguridad tanto Comerciales como Open Source, por tal razón se toma como base las herramientas consideradas las más utilizadas en trabajos realizados [86] [87] [88], así también considerando el top 10 de los riesgos más habituales en aplicaciones web propuesto por OWASP como lo indica la Figura. 29.

OWASP Top 10 – 2013 (Nuevo)
A1 – Inyección
A2 – Pérdida de Autenticación y Gestión de Sesiones
A3 – Secuencia de Comandos en Sitios Cruzados (XSS)
A4 – Referencia Directa Insegura a Objetos
A5 – Configuración de Seguridad Incorrecta
A6 – Exposición de Datos Sensibles
A7 – Ausencia de Control de Acceso a las Funciones
A8 – Falsificación de Peticiones en Sitios Cruzados (CSRF)
A9 – Uso de Componentes con Vulnerabilidades Conocidas
A10 – Redirecciones y reenvíos no validados

Figura 29. Top 10 de los riesgos de seguridad en aplicaciones web. [89]

TABLA XXIX. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS DE SEGURIDAD

HERRAMIENTA	OWASP ZAP	Burp Suite	Snort	WireShark (Sniffer)	Solarwinds	GFI LANguard	Suricata
Dirección url del fabricante	https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project	https://portswigger.net/burp/	https://www.snort.org/	https://www.wireshark.org/	http://www.solarwinds.com/es/	http://www.gfi.com/products-and-solutions/network-security-solutions/gfi-languard	https://suricata-ids.org/
Funcionalidad	Ataques de penetración hacia sistemas web (disciplina conocida como Pentesting)	Ataques de penetración hacia sistemas web	Sistema de Detección y Prevención de intrusos	Analizador de protocolos de red	Detección, supervisión y gestión de redes	Escáner de redes y administrador de parches	Sistema de Detección y Prevención de intrusos
Comercial / Open Source	Open Source	Comercial / Open Source	Open Source	Open Source	Comercial	Comercial	Open Source
Plataforma de funcionamiento	Windows, Linux, MacOS	Mac, Linux, Windows	Windows, Linux	Windows, Linux, MacOS, Solaris, Android, etc.	Windows	Windows	Linux, FreeBSD, OpenBSD, MacOS, Windows.
Requisitos	Java 1.7 o superior	Java 1.5 o superior	Archivo WinPcap	Librería pcap	No se evidencia	Microsoft .NET Framework 4.5.1	Compiladores como: gcc, g++, etc. Librerías como: libcap, libz, etc.

Idiomas soportados	Más de 12 idiomas entre ellos el español	Ingles	Ingles	Ingles	Ingles	Ingles	Ingles
Manual de instalación	No se evidencia	https://support.portswigger.net/customer/en/portal/topics/718317-installing-and-configuring-burp/articles	https://www.snort.org/documents#OfficialDocumentation	https://www.wreshark.org/download/docs/developer-guide-a4.pdf	No se evidencia	http://manuals.gfi.com/es/languard/content/acm/topics/installing_gfi_languard/introduction_topic.htm?cshid=1001	https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Suricata_Installation
Manual de usuario	https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents	https://portswigger.net/burp/help/contents.html	http://manual-snort-org.s3-website-us-east-1.amazonaws.com/	https://www.wreshark.org/download/docs/user-guide-a4.pdf	No se evidencia	http://www.gfi.com/documents/smartguides/lanss_smartguide_en.pdf	http://suricata.readthedocs.io/en/latest/
Soporte	Twitter, Blog, Videos, grupo de Usuarios	Comunidad, centro de ayuda, base de conocimiento	Blog, informe de errores	Videos, presentaciones, wiki, preguntas en línea	Foros, capacitación, artículos de la base de conocimientos, mantenimiento	Comunidad, videos, libros,	Comunidad, correo.

- **Valoración numérica de herramientas de Seguridad**

TABLA XXX. SISTEMA OPERATIVO

	Windows	Linux	MacOS	TOTAL
OWASP ZAP	✓	✓	✓	3
Burp suite	✓	✓	✓	3
Snort	✓	✓	X	2
WireShark	✓	✓	✓	3
Solarwinds	✓	X	X	1
GFI LANguard	✓	X	X	1
Suricata	✓	✓	✓	3

TABLA XXXI. SOPORTE Y DOCUMENTACIÓN

	Manual de Instalación	Manual de usuario	Foros	Comunidad	TOTAL
OWASP ZAP	X	✓	X	✓	2
Burp suite	✓	✓	X	X	2
Snort	✓	✓	X	✓	3
WireShark	✓	✓	X	X	2
Solarwinds	X	X	✓	X	1
GFI LANguard	✓	✓	X	✓	3
Suricata	✓	✓	X	✓	3

TABLA XXXII. SOPORTE DE PRUEBAS

	Monitoreo/ tiempo real	Multihilo	Peticiones HTTPS	Complejidad	Proxy	Alertas instantáneas	TOTAL
OWASP ZAP	✓	✓	✓	Intermedio	✓	✓	5
Burp suite	✓	✓	X	Avanzado	✓	X	3
Snort	✓	✓	X	Medio	✓	✓	4
WireShark	✓	X	X	Medio	X	✓	2
Solarwinds	✓	✓	✓	Avanzado	✓	✓	5
GFI LANguard	✓	X	✓	Medio	✓	✓	4
Suricata	✓	✓	X	Intermedio	X	✓	3

TABLA XXXIII. RESULTADOS

	Sistema Operativo	Soporte y Documentación	Soporte de pruebas	TOTAL
OWASP ZAP	3	2	5	10
Burp suite	3	2	3	8
Snort	2	3	4	9
Suricata	3	3	3	9
WireShark	3	2	2	7
Solarwinds	1	1	5	7
GFI LANguard	1	3	4	8

Del conjunto de herramientas evaluadas sobresalen **OWASP ZAP**, **Snort** y **Suricata**, la primera herramienta diseñada para pentesting, cubre la mayoría de riesgos considerados los más habituales en los últimos años como: inyección SQL, pérdida de autenticación, referencia directa insegura a objetos, entre otros.

La segunda y tercera herramienta las mismas que son Open Source compatible para Linux y Windows, constan con la mayoría de aspectos positivos; **Snort** puede ser configurada para trabajar en tres modos distintos como *Modo de detector*, *Modo de Registrador de paquetes* o *Modo de Intrusion Detection System*, es fácilmente configurable y cuenta con soporte en cuanto a documentación, por su parte **Suricata** tiene la ventaja que es Multihilo y básicamente tiene las mismas funcionalidades de Snort pero con poca documentación disponible dificultando su entendimiento.

A pesar que ninguna de las herramientas antes revisadas se acopla a las pruebas de seguridad a nivel de aplicación **se propone OWASP ZAP** y **Snort como las más idóneas para pruebas futuras en caso de querer realizar pruebas de seguridad de gestión de redes o para detección de intrusos basado en red**. Cabe recalcar que con este análisis no se pretende forzar su utilización, debido a que hay una infinidad de herramientas para diversos ataques, simplemente es una visión objetiva de este trabajo.

- **Herramientas para pruebas de rendimiento**

Las pruebas de rendimiento tienen como objetivo simular el acceso e interacción de una determinada cantidad de usuarios con el sistema, para evaluar el comportamiento del mismo y determinar la velocidad de respuesta de la aplicación. La elección de una herramienta para pruebas de rendimiento depende de muchos factores específicamente

guiados al software a probar por ello se pretende aumentar criterios de evaluación para este segmento de herramientas realizando una breve descripción de los mismos:

- **Visualización en tiempo real:** permite a la herramienta asignar una IP a un usuario virtual para acercar la prueba a la realidad de las peticiones concurrentes de usuarios conectados en distintas estaciones de trabajo.
- **Soporte IP Spoofing:** permite a la herramienta de pruebas asignar una IP a un usuario virtual para acercar la prueba a la realidad de las peticiones concurrentes de usuarios conectados en distintas estaciones de trabajo.
- **Pruebas programadas:** permite programar el inicio, las paradas y el fin de un plan de pruebas específicas.
- **Proxy http:** para que las pruebas se realicen de forma automática, en vez de generar el script con los pasos que se deben seguir para la prueba, se utiliza esta funcionalidad que hace interfaz con el navegador Web normal y graba las acciones que se realizan en él, generando las instrucciones o script necesarios para realizar la prueba específica.
- **Proxy https:** tiene el mismo objetivo y filosofía del proxy HTTP, solo que este graba de forma especial las secuencias, debido a que debe codificar y decodificar las peticiones y respuestas y usa constantemente las key y certificados propios de SSL dependiendo de la versión y del cifrado utilizado.
- **Protocolos:** elementos de comunicación para capturar, manipular y emular peticiones. Entre ellos, los más comunes son HTTP 1.0/1.1 HTTPS (SSL). Es uno de los requisitos básicos de las herramientas y se considera una ventaja comparativa el hecho de que posea la funcionalidad de proxy HTTP o HTTPS. [90]

De acuerdo a trabajos relevantes realizados de comparación de herramientas para pruebas de rendimiento [90] [91] [92] se realiza la siguiente comparativa:

TABLA XXXIV. COMPARATIVA DE HERRAMIENTAS PARA PRUEBAS DE RENDIMIENTO

HERRAMIENTA	FunkLoad	NeoLoad	Jmeter	Gatling	Qtest 5.0	MultiMechanize
Dirección url del fabricante	https://pypi.python.org/pypi/funkload/	http://www.neotys.com/neoload/overview	http://jmeter.apache.org/	http://gatling.io/#/	http://www.qotium.com/	https://multi-mechanize.readthedocs.io/en/latest/
Comercial / Open Source	Open Source	Comercial	Open Source	Open Source	Comercial	Open Source
Plataforma de funcionamiento	Linux, Windows	Windows, Solaris 10	Cualquier plataforma que soporte Java 1.4.2 o posterior	Linux, Mac Windows	Windows NT/2000/2003/XP	Linux, Windows
Visualización en tiempo real	NO	SI	SI	SI	SI	NO
Requisitos	Python 2.7	No se evidencia	JDK 4.0 o superior	JDK 6.0 o superior	No se evidencia	Python 2.6 o 2.7
Idiomas soportados	Inglés	Inglés, Francés	Español, Inglés, Japonés, Noruego Alemán, Francés, Chino Polaco, Turco.	Inglés	Inglés	Inglés

Manual de instalación	https://funkload.nuxeo.org/installation.html	http://www.neotys.com/documents/document/neoload/latest/en/html/#617.htm	http://jmeter.apache.org/usermanual/get-started.html#install	No se evidencia	No se evidencia	https://github.com/cgoldberg/multi-mechanize
Manual de usuario	https://funkload.nuxeo.org/	http://www.neotys.com/documents/document/neoload/latest/en/html/#1755.htm	http://jmeter.apache.org/usermanual/index.html	http://gatling.io/docs/2.2.3/#user-s-guide	No se evidencia	No se evidencia
Protocolos	HTTP, HTTPS, TCP, LDAP	HTTP 1.0-1.1- HTTPS	HTTP, HTTPS, SOAP, JDBC, LDAP, JMS, TCP, SMTP, POP3, IMAP.	HTTP, HTTPS, JDBC, JMS, LDAP, POP, IMAP.	HTTP 1.0-1.1- HTTPS	HTTP
Soporte IP Spoofing	NO	SI	SI	NO	SI	NO
Pruebas programadas	SI	SI	SI	SI	SI	NO
Proxy http	SI	SI	SI	SI	SI	SI
Proxy https	NO	SI	SI	SI	SI	NO
Soporte	Comunidad	Comunidad, soporte de contacto, email, redes sociales, tutoriales	Tutoriales en PDF, comunidad, acceso a gitHub (todo tipo de información de JMeter)	Comunidad, Blog, servicios de asistencia y consultoría	No se evidencia	Comunidad

- Valoración numérica de herramientas para Rendimiento

TABLA XXXV. SISTEMA OPERATIVO

Sistema Operativo				
	Windows	Linux	MacOS	TOTAL
FunkLoad	✓	✓	X	2
NeoLoad	✓	X	X	1
Jmeter	✓	✓	✓	3
Gatling	✓	✓	✓	3
Qtest 5.0	✓	X	X	1
MultiMechanize	✓	✓	X	2

TABLA XXXVI. SOPORTE Y DOCUMENTACIÓN

	Manual de Instalación	Manual de usuario	Foros	Comunidad	TOTAL
FunkLoad	✓	✓	X	✓	3
NeoLoad	✓	✓	X	✓	3
Jmeter	✓	✓	X	✓	3
Gatling	X	✓	X	✓	2
Qtest 5.0	X	X	X	X	0
MultiMechanize	✓	X	X	✓	2

TABLA XXXVII. PROTOCOLOS SOPORTADOS

	HTTP	HTTPS	TCP	LDAP	SMTP	TOTAL
FunkLoad	✓	✓	✓	✓	X	4
NeoLoad	✓	✓	X	X	X	2
Jmeter	✓	✓	✓	✓	✓	5
Gatling	✓	✓	X	✓	X	3
Qtest 5.0	✓	✓	X	X	X	2
MultiMechanize	✓	X	X	X	X	1

TABLA XXXVIII. REQUISITOS MINIMOS DEL MÓDULO A PROBAR

	Open Source	Visualización / tiempo real	TOTAL
FunkLoad	✓	X	1
NeoLoad	X	✓	1
Jmeter	✓	✓	2
Gatling	✓	✓	2
Qtest 5.0	X	✓	1
MultiMechaniz e	✓	X	1

TABLA XXXIX. RESULTADOS

	Sistema Operativo	Soporte y Documentación	Protocolos soportados	Requisitos del módulo a probar	TOTAL
Jmeter	3	3	5	1	12
MultiMechanize	2	2	1	1	6
FunkLoad	2	3	4	2	11
Gatling	3	2	3	2	10
NeoLoad	1	3	2	1	7
Qtest 5.0	1	0	2	1	4

De acuerdo a los lineamientos establecidos del software a probar el mismo que trabaja con el protocolo HTTPS se evidencia del conjunto de herramientas evaluadas que las que poseen la mayoría de características positivas son **NeoLoad** y **Jmeter**.

NeoLoad es una herramienta constando con manual de instalación y de usuario para mayor entendimiento del Tester, además se visualiza en tiempo real las pruebas lo que es muy importante para saber qué decisión tomar en el momento. Por su parte rigiéndonos a los lineamientos de la UTI los que trabajan bajo Linux la herramienta **más completa se puede evidenciar que es Jmeter** la misma que soporta cualquier plataforma y consta con toda la información requerida para su entendimiento visualizando en tiempo real las pruebas. **Llegando a la conclusión que para estas instancias se recomienda Jmeter como herramienta para pruebas de rendimiento.**

FASE 3-4: ACTUACIÓN - APRENDIZAJE

Versión	Fecha	Autor
1.0	08/01/2017	Lady Iñiguez Carchi

Objetivos

Los objetivos de esta fase son:

- Validar la propuesta con los integrantes del departamento de desarrollo de software de la UTI resolviendo las no conformidades.
- Documentar todo el proceso de validación.

Técnicas

- Poner en práctica las plantillas levantadas y herramientas de pruebas definidas.

Involucrados

- Miembros del GDS.
- Director de Trabajo de Titulación.
- Tesista.

Problemas y Soluciones

Problema 1: Poco conocimiento del uso de las herramientas especificadas

Solución 1: Investigar su funcionalidad y ver videos tutoriales de los mismos.

Glosario de Términos

CMMI, Capability Maturity Model Integration, Modelo de Capacidad de Madurez Integrada

PPQA, Aseguramiento de la Calidad del Proceso y del Producto.

6.4. FASE ACTUACIÓN - APRENDIZAJE: Validación de la propuesta.

6.4.1. Introducción

En esta fase se valida la propuesta de mejora de procesos basado en el Modelo de Capacidad de Madurez Integrada CMMI v1.3 específicamente en el área de Aseguramiento de la Calidad del Proceso y del Producto (PPQA).

6.4.2. Validación de plantillas.

Se asignó por parte del Director de la Unidad de Telecomunicaciones e Información y del Subdirector de Desarrollo de Software, como piloto de pruebas el sistema en desarrollo “Sistema de Información Académico Administrativo Financiero” (SIAAF) de la Universidad Nacional de Loja, específicamente el Módulo de Planificación y Desarrollo Institucional. **(Anexo 9)**

Las plantillas han sido validadas de acuerdo al orden de los procesos de Aseguramiento de la Calidad del Proceso y del Producto establecidos en la fase “Establecer”, cada una de estas plantillas han sido revisadas y aprobadas por la Ing. Lissette López Analista/Desarrolladora de la Unidad de Telecomunicaciones e Información.



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	11/11/2015

NOMBRE DE DOCUMENTO	Plan de Aseguramiento de la Calidad del Proceso y del Producto
VERSIÓN	1.0

ELABORADO POR:	Ing. Lissette López – Lady Iñiguez Líder de PPQA	
	FECHA:	05/12/2016
REVISADO POR:	Ing. Edison Coronel Líder de Proyecto	
	Ing. Lissette López Líder de PPQA	
	FECHA:	06/12/2016
APROBADO POR:	Ing. Edison Coronel Líder de Proyecto	
	FECHA:	06/12/2016

Contenido

1. PROPÓSITO
2. ALCANCE
3. ROLES Y RESPONSABILIDADES
4. PRODUCTOS A EVALUAR Y ENTREGABLES
5. PLANIFICAR ACTIVIDADES DE QA

1. PROPÓSITO

El propósito de este Plan es especificar las actividades que se realizarán para asegurar la calidad del Módulo de Planificación y Desarrollo Institucional estableciendo roles y responsabilidades a las personas quienes estarán inmiscuidas en el desarrollo del proyecto, así como determinando productos a validar y planificando fechas para el cumplimiento de cada fase del módulo en cuestión.

2. ALCANCE

Se puntualiza las iteraciones que obligatoriamente se tendrá que llevar a cabo para el desarrollo del Módulo de Planificación y Desarrollo Institucional, generando o utilizando documentación que aportará información durante el proceso de desarrollo.

3. ROLES Y RESPONSABILIDADES

Se describirá roles y responsabilidades de las personas que serán encargados del aseguramiento de la calidad del proceso y del producto durante el desarrollo del proyecto.

Rol	Nombre	Responsabilidad
Líder de Proyecto	Ing. Edison Coronel	Responsable de tomar decisiones adecuadas para las acciones a realizar durante el desarrollo del proyecto.
Líder de PPQA	Ing. Lissette López	Responsable de monitorear el progreso del aseguramiento de la calidad del proceso y del producto en las fases del proyecto.
Analista	Ing. Jorge Malla	Encargado de diseñar los productos de trabajo que se obtenga en la fase de

		Análisis y Diseño del proyecto.
Desarrollador	Ing. Jorge Malla	Encargado de codificar el modelo de clases y las funcionalidades del software.
Tester	Lady Iñiguez	Responsable de validar los productos de trabajo obtenidos en la fase de Análisis y Diseño así como también, realizar el control de calidad aplicando pruebas pertinentes que se ajusten al proyecto.

4. PRODUCTOS A EVALUAR Y ENTREGABLES

Se identificarán los productos de trabajo sobre los cuales se va a trabajar durante el ciclo de vida del proyecto, el responsable que los evaluará y el entregable asociado luego de evaluar dichos documentos.

Producto	Evaluador	Responsable	Entregable asociado
Especificación de Requisitos de Software (ERS)	Tester	Lady Iñiguez	<ul style="list-style-type: none"> • Plantilla Checklist de validación de requisitos. • Informe de no conformidades de validación de requisitos.
Prototipo	Tester	Lady Iñiguez	<ul style="list-style-type: none"> • Plantilla Checklist de validación de prototipo.

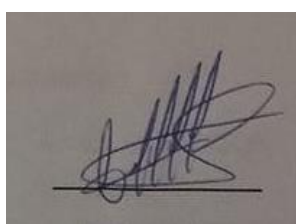
			<ul style="list-style-type: none"> • Informe de no conformidades de validación de prototipo.
Casos de Uso (CU)	Tester	Lady Iñiguez	<ul style="list-style-type: none"> • Plantilla Checklist de validación de casos de uso. • Informe de no conformidades de validación de casos de uso
Plan de pruebas	Líder de PPQA	Ing. Lissette López Lady Iñiguez	<ul style="list-style-type: none"> • Plan de pruebas. • Plantilla informe de errores. • Plantilla informe de aprobación.

5. PLANIFICAR ACTIVIDADES DE QA

Se enumeraran las actividades para realizar el Aseguramiento de la Calidad del Proceso y del Producto (PPQA) y se definirán fechas tentativas para el cumplimiento de cada una de las actividades.

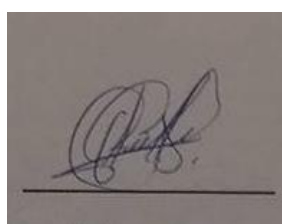
Actividad	Descripción	Fecha inicio	Fecha de fin
Definir Plan de PPQA	Se define el Plan de PPQA indicando responsabilidades, productos de trabajo a evaluar y actividades a realizar.	05/11/2016	06/12/2016
Validar ERS	Se envía el documento ERS a ser validado.	06/12/2016	08/12/2016
Validar Prototipo	Se envía el documento de	08/12/2016	12/12/2016

	prototipo a ser validado.		
Validar CU	Se envía el documento de casos de uso a ser validado.	12/12/2016	14/12/2016
Definir plan de pruebas	Se define el Plan de pruebas donde constaran las pruebas a realizarse para este proyecto en ejecución.	14/12/2016	19/12/2016
Realizar pruebas	Realizar las pruebas definidas en el Plan de pruebas	19/12/2016	30/12/2016



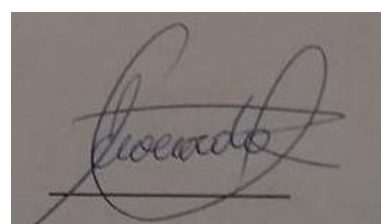
Elaborado

Ing. Lissette López



Elaborado

Lady Iñiguez



Aprobado

Ing. Edison Coronel



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	11/11/2015

NOMBRE DE DOCUMENTO	Checklist de Validación de Requisitos
VERSIÓN	1.0

VALIDADO POR:	Lady Iñiguez Tester	
	FECHA:	06/12/2016
APROBADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	07/12/2016

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. CHECKLIST DE REVISIÓN**
- 4. CONCLUSION**

1. DESCRIPCIÓN

El Checklist de validación de requisitos es planteado para asegurar que todos los requisitos sugeridos por el usuario en la etapa de Planificación para el desarrollo del Módulo de Planificación y Desarrollo Institucional hayan cumplido con los criterios de revisión establecidos.

2. OBJETIVOS

- Validar los requisitos funcionales y no funcionales del Módulo de Planificación y Desarrollo Institucional en base a criterios establecidos en el checklist de validación de requisitos.
- Descubrir no conformidades en caso de existir en la descripción de requisitos, en cuanto a si son correctos, que no exista ambigüedad y que sean concisos y fáciles de entender.

3. CHECKLIST DE REVISIÓN

CRITERIOS DE REVISIÓN	RF001		RF002		RF003		RF004		RF005		RF006		RF007		RF008		RF009		RF010		RF011	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
Los requisitos tienen un código de identificación	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tienen un nombre	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tiene una descripción		λ		λ		λ		λ		λ		λ		λ		λ		λ		λ		λ
Los requisitos están clasificados como requisitos Funcionales y no Funcionales	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son ambiguos		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos son correctos (reflejar una necesidad real)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son completos (proporcionan la información suficiente para su comprensión)		λ		λ		λ		λ		λ		λ		λ		λ		λ		λ		λ
Los requisitos son concisos y fáciles de entender	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos están clasificados por prioridad (Alta o Media)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	

CRITERIOS DE REVISIÓN	RF012		RF013		RF014		RF015		RF016		RNF017		RNF018		RNF019		RNF020		RNF021		
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	
Los requisitos tienen un código de identificación	✓		✓		✓		✓		✓			✗		✗		✗		✗		✗	
Los requisitos tienen un nombre	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos tiene una descripción		✗		✓		✓		✗		✗		✓		✓		✓		✓		✓	
Los requisitos están clasificados como requisitos Funcionales y no Funcionales	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos son ambiguos		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son correctos (reflejar una necesidad real)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos son completos (proporcionan la información suficiente para su comprensión)		✗		✓		✗		✗		✗		✓		✓		✓		✓		✓	
Los requisitos son concisos y fáciles de entender	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos están clasificados por prioridad (Alta o Media)	✓		✓		✓		✓		✓			✗		✗		✗		✓		✓	

4. CONCLUSION

- En la primera versión 1.0 La mayoría de criterios de revisión establecidos no se han cumplido en la descripción de requisitos, existiendo varios cambios por realizar, los criterios que no se cumplen en su mayoría son de descripción, es decir los requisitos no tienen una descripción clara y entendible, llegando a ser difícil de entender.



Validado

Lady Iñiguez



Aprobado

Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	11/11/2015

NOMBRE DE DOCUMENTO	No-Conformidades de Validación de Requisitos
VERSIÓN	1.0

ELABORADO POR:	Lady Iñiguez Tester	
	FECHA:	07/12/2016
REVISADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	07/12/2016

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. DESCRIPCIÓN DE NO CONFORMIDADES**
- 4. RECOMENDACIONES**

1. DESCRIPCIÓN

La plantilla de no conformidades de validación de requisitos se planteó para puntualizar las no conformidades que dieron como resultado en la revisión de los requisitos del Módulo de Planificación y Desarrollo Institucional.

2. OBJETIVOS

- Detallar las no conformidades encontradas en la descripción de requisitos.

3. DESCRIPCIÓN DE NO CONFORMIDADES

- Los códigos de identificación de los requisitos tienen una numeración equivocada. **(se repite dos veces la numeración RF005, no se evidencia la numeración del RF008)**
- Existe faltas de ortografía en la descripción de requisitos. **(RF005, RF007)**
- Los requisitos funcionales no tienen una descripción. **(Todos)**
- Los requisitos funcionales no son completos, no proporcionan la suficiente información para su comprensión. **(Todos)**
- Los requisitos no funcionales, no tienen un código de identificación. **(Todos)**
- Los requisitos no funcionales, no están clasificados por prioridad. **(Todos)**

4. RECOMENDACIONES

- Se tome más precaución a la hora de digitalizar la información pertinente a la descripción de requisitos.
- Se lleve un mismo esquema para la descripción de requisitos funcionales y no funcionales para evitar inconformidades futuras.



Elaborado

Lady Iñiguez



Revisado

Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	11/11/2015

NOMBRE DE DOCUMENTO	Checklist de Validación de Requisitos
VERSIÓN	1.1

VALIDADO POR:	Lady Iñiguez Tester	
	FECHA:	08/12/2016
APROBADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	08/12/2016

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. CHECKLIST DE REVISIÓN**
- 4. CONCLUSION**

1. DESCRIPCIÓN

El Checklist de validación de requisitos es planteado para asegurar que todos los requisitos sugeridos por el usuario en la etapa de Planificación para el desarrollo del Módulo de Planificación y Desarrollo Institucional hayan cumplido con los criterios de revisión establecidos.

5. OBJETIVOS

- Validar los requisitos funcionales y no funcionales del Módulo de Planificación y Desarrollo Institucional en base a criterios establecidos en el checklist de validación de requisitos.
- Descubrir no conformidades en caso de existir en la descripción de requisitos, en cuanto a si son correctos, que no exista ambigüedad y que sean concisos y fáciles de entender.

2. CHECKLIST DE REVISIÓN

CRITERIOS DE REVISIÓN	RF001		RF002		RF003		RF004		RF005		RF006		RF007		RF008		RF009		RF010		RF011	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
Los requisitos tienen un código de identificación	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tienen un nombre	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tiene una descripción	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos están clasificados como requisitos Funcionales y no Funcionales	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son ambiguos		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos son correctos (reflejar una necesidad real)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son completos (proporcionan la información suficiente para su comprensión)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son concisos y fáciles de entender	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos están clasificados por prioridad (Alta o Media)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	

CRITERIOS DE REVISIÓN	RF012		RF013		RF014		RF015		RF016		RNF017		RNF018		RNF019		RNF020		RNF021	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
Los requisitos tienen un código de identificación	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tienen un nombre	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos tiene una descripción	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos están clasificados como requisitos Funcionales y no Funcionales	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son ambiguos		✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Los requisitos son correctos (reflejar una necesidad real)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son completos (proporcionan la información suficiente para su comprensión)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos son concisos y fáciles de entender	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Los requisitos están clasificados por prioridad (Alta o Media)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	

3. CONCLUSION

- En esta segunda versión todos los criterios de revisión establecidos en el checklist de requisitos han sido cumplidos a cabalidad, entregando un documento ordenado y entendible, no existiendo ningún cambio por realizar.



Validado

Lady Iñiguez



Aprobado

Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE FINALIZACIÓN	11/11/2015

NOMBRE DE DOCUMENTO	Checklist de Validación de Prototipos
VERSIÓN	1.0

VALIDADO POR:	Lady Iñiguez Tester	
	FECHA:	<19/12/2016>
APROBADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	<19/12/2016>

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. CHECKLIST DE REVISIÓN**
- 4. CONCLUSION**

1. DESCRIPCIÓN

El checklist de validación de prototipos es planteado para asegurar que los prototipos diseñados en la etapa de Diseño para el Módulo de Planificación y Desarrollo Institucional cumplen con las peticiones del usuario y con los criterios de diseño establecidos.

2. OBJETIVOS

- Validar los prototipos de acuerdo a los criterios de revisión ya establecidos.
- Descubrir no conformidades en caso de existir en el diseño de prototipos, en lo que tiene que ver a mensajes de información para el usuario, que los colores tengan contraste y que la ortografía de las mismas este correcta.

3. CHECKLIST DE REVISIÓN

CRITERIOS DE REVISIÓN	PROT1		PROT2		PROT3		PROT4		PROT5		PROT6		PROT7		PROT8		PROT9		
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	
El prototipo tienen un código de identificación	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
La pantalla tienen un título que la identifique	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
En caso de existir los mensajes de información presentados están situados en una parte de la pantalla donde no interfieren con otros elementos.																			
El tamaño de la fuente en toda la pantalla es legible a distancia	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
El color de la fuente da contraste con el fondo de la pantalla.	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Todos los componentes que conforman la pantalla están alineados de forma organizada (botones, input, etiquetas)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
La ortografía de los siguientes elementos es correcta																			
Botones			✓											✓		✓			
Etiquetas (labels, mensajes informativos...)	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓
Input			✓				✓				✓		✓		✓				

CRITERIOS DE REVISIÓN	PROT10		PROT11		PROT12		PROT13		PROT14		PROT15		PROT16		PROT17		PROT18	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
El prototipo tienen un código de identificación	✓		✓		✓		✓		✓		✓		✓		✓		✓	
La pantalla tienen un título que la identifique	✓		✓		✓		✓		✓		✓		✓		✓		✓	
En caso de existir los mensajes de información presentados están situados en una parte de la pantalla donde no interfieren con otros elementos.																		
El tamaño de la fuente en toda la pantalla es legible a distancia	✓		✓		✓		✓		✓		✓		✓		✓		✓	
El color de la fuente da contraste con el fondo de la pantalla.	✓		✓		✓		✓		✓		✓		✓		✓		✓	
Todos los componentes que conforman la pantalla están alineados de forma organizada (botones, input, etiquetas)	✓		✓		✓		✓		✓		✓		✓		✓		✓	
La ortografía de los siguientes elementos es correcta																		
Botones			✓		✓						✓		✓					
Etiquetas (labels, mensajes informativos...)	✓		✓		✓		✓		✓		✓		✓		✓		✓	
Input	✓		✓		✓		✓		✓		✓		✓				✓	

CRITERIOS DE REVISIÓN	PROT19		PROT20		PROT21		PROT22		PROT23		PROT24	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
El prototipo tienen un código de identificación	✓		✓		✓		✓		✓		✓	
La pantalla tienen un título que la identifique	✓		✓		✓		✓		✓		✓	
En caso de existir los mensajes de información presentados están situados en una parte de la pantalla donde no interfieren con otros elementos.												
El tamaño de la fuente en toda la pantalla es legible a distancia	✓		✓		✓		✓		✓		✓	
El color de la fuente da contraste con el fondo de la pantalla.	✓		✓		✓		✓		✓		✓	
Todos los componentes que conforman la pantalla están alineados de forma organizada (botones, input, etiquetas)	✓		✓		✓		✓		✓		✓	
La ortografía de los siguientes elementos es correcta												
Botones	✓		✓						✓		✓	
Etiquetas (labels, mensajes informativos...)	✓		✓		✓		✓		✓		✓	
Input	✓		✓				✓		✓		✓	

4. CONCLUSION

- Todos los criterios de revisión establecidos han sido cumplidos a cabalidad, sin encontrarse ninguna inconformidad, para poder continuar con la siguiente fase, quedando por asentado que en esta primera versión todos los prototipos de pantalla pasaron la revisión.



Validado
Lady Iñiguez



Aprobado
Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	Checklist de Validación de Casos de Uso
VERSIÓN	1.0

VALIDADO POR:	Lady Iñiguez Tester	
	FECHA:	<19/12/2016>
APROBADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	<20/12/2016>

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. CHECKLIST DE REVISIÓN**
- 4. CONCLUSION**

1. DESCRIPCIÓN

El checklist de validación de casos de uso es planteado para asegurar que los casos de uso tengan referencia a los requisitos y prototipos planteados en el desarrollo del Módulo de Planificación y Desarrollo Institucional, además de cumplir con los criterios de revisión establecidos.

2. OBJETIVOS

- Validar los casos de uso de acuerdo a los criterios de revisión ya establecidos.
- Descubrir no conformidades en caso de existir en la evaluación de casos de uso.

3. CHECKLIST DE REVISIÓN

CRITERIOS DE REVISIÓN	CU001		CU002		CU003		CU004		CU005		CU006	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
Los casos de uso tiene un código de identificación	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un nombre	✓		✓		✓		✓		✓		✓	
El nombre refleja algo que el sistema debe hacer	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen una breve descripción	✓		✓		✓		✓		✓		✓	
La descripción es clara	✓		✓		✓		✓		✓		✓	
Los casos de uso hacen referencia a los requisitos asociados	✓		✓		✓		✓		✓		✓	
Los casos de uso hacen referencia al prototipo asociado		x		x		x		x		x		x
Los casos de uso tienen una precondición	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un postcondición	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un flujo principal de eventos	✓		✓		✓		✓		✓		✓	
Se contempla todos los posibles flujos alternos		x		x		x		x		x		x
Los pasos de los flujos tiene una secuencia lógica		x		x		x		x		x		x

4. CONCLUSION

- Existen algunos criterios de revisión importantes en la descripción de casos de uso que no se cumplen a cabalidad, como por ejemplo que los casos de uso no hacen referencia la prototipo designado, o que no se contempla todos los flujos alternos y los que si se contemplan no tienen una secuencia lógica, existiendo varios cambios por realizar.

A rectangular box containing a handwritten signature in blue ink on a grey background. The signature is stylized and appears to be 'Lady Iñiguez'.

Validado
Lady Iñiguez

A rectangular box containing a handwritten signature in blue ink on a grey background. The signature is stylized and appears to be 'Ing. Lissette López'.

Aprobado
Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	No-Conformidades de Validación de Casos de Uso
VERSIÓN	1.0

ELABORADO POR:	Lady Iñiguez Tester
	FECHA:
REVISADO POR:	Ing. Lissette López Líder de PPQA
	FECHA:

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. DESCRIPCIÓN DE NO CONFORMIDADES**
- 4. RECOMENDACIONES**

1. DESCRIPCIÓN

La plantilla de no conformidades de validación de casos de uso se planteó para puntualizar las no conformidades que dieron como resultado en la revisión de los casos de uso del Módulo de Planificación y Desarrollo Institucional.

2. OBJETIVOS

- Detallar las no conformidades encontradas en la descripción de requisitos.

3. DESCRIPCIÓN DE NO CONFORMIDADES

- Los casos de uso hacen referencia a prototipos que no tiene relación al mismo. **(Todos)**
- De acuerdo al diseño de las pantallas no se considera todos los posibles flujos alternos. **(CU006)**
- De acuerdo al diseño de las pantallas existen flujos de pasos que no tienen una secuencia lógica. **(Todos)**
- Existe un caso de uso que no hace relación a ninguna pantalla. **(CU003)**

4. RECOMENDACIONES

- Se tome como referencia el prototipo aprobado de pantallas para diseñar los casos de uso, permitiendo tener una relación coherente entre las pantallas y la descripción de casos de uso evitando confusiones futuras.
- Se tenga más precaución a la hora de diseñar los casos de uso describiendo todos los flujos alternos posibles para evitar futuras inconformidades.



Elaborado

Lady Iñiguez



Revisado

Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	Checklist de Validación de Casos de Uso
VERSIÓN	1.1

VALIDADO POR:	Lady Iñiguez Tester	
	FECHA:	<20/12/2016>
APROBADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	<20/12/2016>

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. CHECKLIST DE REVISIÓN**
- 4. CONCLUSION**

1. DESCRIPCIÓN

El checklist de validación de casos de uso es planteado para asegurar que los casos de uso tengan referencia a los requisitos y prototipos planteados en el desarrollo del Módulo de Planificación y Desarrollo Institucional, además de cumplir con los criterios de revisión establecidos.

2. OBJETIVOS

- Validar los casos de uso de acuerdo a los criterios de revisión ya establecidos.
- Descubrir no conformidades en caso de existir en la evaluación de casos de uso, que van desde la lógica de sus flujos hasta que reflejen realmente lo que el software debe hacer.

3. CHECKLIST DE REVISIÓN

CRITERIOS DE REVISIÓN	CU001		CU002		CU003		CU004		CU005		CU006	
	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO	SI	NO
Los casos de uso tiene un código de identificación	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un nombre	✓		✓		✓		✓		✓		✓	
El nombre refleja algo que el sistema debe hacer	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen una breve descripción	✓		✓		✓		✓		✓		✓	
La descripción es clara	✓		✓		✓		✓		✓		✓	
Los casos de uso hacen referencia a los requisitos asociados	✓		✓		✓		✓		✓		✓	
Los casos de uso hacen referencia al prototipo asociado	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen una precondición	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un postcondición	✓		✓		✓		✓		✓		✓	
Los casos de uso tienen un flujo principal de eventos	✓		✓		✓		✓		✓		✓	
Se contempla todos los posibles flujos alternos	✓		✓		✓		✓		✓		✓	
Los pasos de los flujos tiene una secuencia lógica	✓		✓		✓		✓		✓		✓	

4. CONCLUSION

- En esta segunda versión se llega a la conclusión que se ha cumplido con todos los criterios de revisión establecidos en la descripción de casos de uso, no existiendo ningún cambio por realizar, obteniendo un documento ordenado y entendible.

A square image showing a handwritten signature in blue ink on a grey background. The signature is stylized and appears to be 'Lady Iñiguez'. Below the signature is a horizontal line.

Validado
Lady Iñiguez

A square image showing a handwritten signature in blue ink on a grey background. The signature is stylized and appears to be 'Lissette López'. Below the signature is a horizontal line.

Aprobado
Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	Plan de Pruebas
VERSIÓN	1.0

ELABORADO POR:	Lady Iñiguez Tester
	FECHA:
APROBADO POR:	Ing. Lissette López Líder de PPQA
	FECHA:

Contenido

1. PROPÓSITO
2. ALCANCE
3. DOCUMENTOS ASOCIADOS
4. ENTREGABLES
5. TIPOS DE PRUEBAS A REALIZAR
6. HERRAMIENTAS A UTILIZAR
7. CASOS DE PRUEBAS
8. GLOSARIO

1. PROPÓSITO

El propósito del Plan de pruebas es determinar si el Módulo de Planificación y Desarrollo Institucional cumple a cabalidad con todos los requerimientos propuestos por el usuario y si cuenta con la calidad esperada. Además de:

- Determinar qué tipos de pruebas se va a realizar.
- Definir la documentación utilizada para la ejecución de pruebas y que información se entregará a la culminación de las mismas.
- Identificar que herramientas se utilizará de acuerdo al tipo de prueba a realizar.
- Elaborar los casos de prueba que permitan descubrir errores en las funcionalidades a ser probadas.

2. ALCANCE

El alcance del presente documento es identificar errores en el Módulo de Planificación y Desarrollo Institucional con el fin de poderlos corregir y poder dar paso a producción.

3. TIPOS DE PRUEBAS A REALIZAR

Se detalla los 3 tipos de pruebas seleccionadas:

- a. **Prueba Funcional:** se centran en las funciones, entradas y salidas, verifican la funcionalidad de un sistema, subsistema o componente. Esta prueba se realiza mediante la técnica de caja negra.

Se trabajará con la partición de equivalencia y casos de uso; ya que al combinarlas los casos de uso darán las interacciones entre el actor y el sistema mientras que la partición de equivalencia permite aplicar valores validos e inválidos.

- b. **Prueba de Seguridad a nivel de aplicación:** aseguran que el software o aplicación incorpore controles apropiados, es decir que un actor solo pueda acceder a las funciones y datos que su rol le tiene permitido.

La técnica que **se utiliza es identificar el tipo de usuario y las funcionalidades a las que tienen acceso dicho usuario:** se crea casos de

prueba para cada tipo de usuario y se verifica que los permisos de acceso funcionen como se lo espera.

c. Prueba de rendimiento

El objetivo es determinar si el usuario estará satisfecho con la velocidad de la aplicación.

Haciendo uso de una herramienta que permita simular el acceso aleatorio de usuarios y determinar tiempos de respuestas.

4. DOCUMENTOS ASOCIADOS

La documentación que se utilizará para la creación de casos de prueba en estas instancias es el documento de Casos de Uso validado y el ERS validado.

5. ENTREGABLES

Los entregables que resultan de la realización de las pruebas son:

- Informe de errores (en caso de evidenciarse errores en la ejecución de pruebas del software).
- Informe de aprobación.

6. HERRAMIENTAS A UTILIZAR

Las herramientas seleccionadas son: (Ver Sección 3.6)

a. Testlink

TestLink es una herramienta Open Source que permite crear y gestionar casos de pruebas y organizarlos en planes de prueba. Estos planes permiten a los miembros del equipo ejecutar casos de test y registrar los resultados dinámicamente, generar informes, mantener la trazabilidad con los requerimientos, así como priorizar y asignar tareas. Trabaja con bases de datos como MySQL, PostgreSQL, MS-SQL.

b. Jmeter

Jmeter es una herramienta de código abierto multiplataforma, diseñado para realizar pruebas de rendimiento. Jmeter nos permite realizar simulaciones de gran carga en el servidor, red o aplicación para comprobar su “fuerza” y para analizar el rendimiento ante diferentes tipos de sobrecarga.

7. CASOS DE PRUEBA

Los casos de prueba que se detallan corresponden a las pruebas de seguridad a nivel de aplicación, debido a que tanto en pruebas de funcionalidad y rendimiento se hará uso de herramientas.

REGISTRO DE CASOS DE PRUEBA				
Módulo de Planificación y Desarrollo Institucional				
Identificador de caso de rol	Versión	Tipo de prueba	Fecha de creación	Fecha de culminación
Administrador de PDI	V1.0	Seguridad a nivel de Aplicación	28/12/2016	28/12/2016
Funciones		Acciones autorizados		Resultado obtenido
Administrar PEDI		Crear PEDI		Exitoso
		Editar PEDI		Exitoso
		Buscar PEDI		Exitoso
		Eliminar PEDI		Exitoso
Administrar Objetivos Operativos		Crear Objetivos Operativos		Exitoso
		Editar Objetivos Operativos		Exitoso
		Buscar Objetivos Operativos		Exitoso
		Eliminar Objetivos Operativos		Exitoso
Administrar Acciones		Crear Acciones		Exitoso
		Editar Acciones		Exitoso
		Buscar Acciones		Exitoso
		Eliminar Acciones		Exitoso
Administrar Indicadores		Crear Acciones		Exitoso
		Editar Indicadores		Exitoso
		Buscar Indicadores		Exitoso
		Eliminar Indicadores		Exitoso
Administrar Metas		Crear Metas		Exitoso
		Editar Metas		Exitoso
		Buscar Metas		Exitoso
		Eliminar Metas		Exitoso

8. GLOSARIO

- **Caja negra**, esta técnica se centran en los resultados obtenidos con datos de entrada válidos y no válidos, sin preocuparse de lo que pase internamente en el sistema.
- **Partición de equivalencia**, se aplican datos válidos o valores que deben aceptarse y datos no validos o valores que deben rechazarse.
- **Open Source**, es el término con el que se conoce al software distribuido y desarrollado libremente.
- **Multiplataforma**, Que puede utilizarse en diversos entornos o sistemas operativos.



Elaborado

Lady Iñiguez



Aprobado

Ing. Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	Informe de errores
VERSIÓN	1.0

ELABORADO POR:	Lady Iñiguez Tester	
	FECHA:	29/12/2016
REVISADO POR:	Lissette López Líder de PPQA	
	FECHA:	29/12/2016

Contenido

- 1. DESCRIPCIÓN**
- 2. OBJETIVOS**
- 3. REPORTE DE ERRORES**

1. DESCRIPCIÓN

El informe de errores es planteado para describir los errores encontrados en la ejecución de los casos de prueba del Módulo de Planificación y Desarrollo Institucional.

2. OBJETIVOS

- Documentar los errores encontrados en el software, para que puedan ser corregidos por el desarrollador encargado del mismo y así poder volver a realizar una próxima evaluación y determinar si el modulo no tiene errores y pueda pasar a producción.

3. REPORTE DE ERRORES

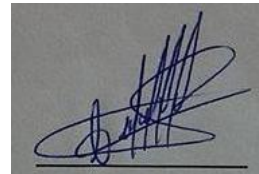
Reporte de prueba Nro: 1			
Fecha de prueba: 29/12/2016		Tipo de prueba: Funcional	
Id. Caso de prueba	Pasos	Nivel de Severidad	Descripción
CU001- <u>MPDI-1: Gestionar PEDI [1]</u>	2. Realiza una búsqueda de los Planes Estratégicos Institucionales por criterios de búsqueda (Código).	Alto	No realiza la acción buscar de un PEDI.
	3. Realiza una búsqueda de los Planes Estratégicos Institucionales por criterios de búsqueda (Nombre).	Alto	No realiza la acción buscar de un PEDI.
	4. Realiza una búsqueda de los Planes Estratégicos Institucionales por criterios de búsqueda erróneos (Código, Nombre).	Alto	No realiza la acción buscar de un PEDI.
	8. Llena los campos obligatorios del PEDI y presiona el botón [Guardar].	Alto	Luego de presionar el botón [Guardar] redirecciona a la interfaz de Metas.
CU003- <u>MPDI-5: Gestionar OO [1]</u>	11. Confirmar la eliminación.	Alto	No actualiza la tabla de Objetivos Operativos, se debe refrescar para que se actualice la tabla.
CU004- <u>MPDI-6: Gestionar Acciones [1]</u>	10. Confirmar la eliminación.	Alto	No actualiza la tabla de Objetivos Operativos, se debe refrescar para que se actualice la tabla.

CU005- MPDI-7: Gestionar <u>Indicadores [1]</u>	10. Confirmar la eliminación.	Alto	No actualiza la tabla de Objetivos Operativos, se debe refrescar para que se actualice la tabla.
CU006- MPDI-8: Gestionar <u>Metas [1]</u>	10. Confirmar la eliminación.	Alto	No actualiza la tabla de Objetivos Operativos, se debe refrescar para que se actualice la tabla.



Elaborado

Lady Iñiguez



Revisado

Lissette López



UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3)
Aseguramiento de la Calidad del Proceso y del Producto (PPQA)

PROYECTO	Módulo de Planificación y Desarrollo Institucional
LÍDER DE PROYECTO	Ing. Edison Coronel
FECHA DE INICIALIZACIÓN	01/11/2015

NOMBRE DE DOCUMENTO	Informe de aprobación
VERSIÓN	1.0

ELABORADO POR:	Lady Iñiguez Tester	
	FECHA:	<04/01/2017>
REVISADO POR:	Ing. Lissette López Líder de PPQA	
	FECHA:	<04/01/2017>

Contenido

- 1. PROPÓSITO**
- 2. REPORTE DE APROBACIÓN**

1. PROPÓSITO

El propósito del reporte de aprobación es puntualizar que el Módulo de Planificación y Desarrollo Institucional cuenta con la calidad esperada y sobretodo precisar que pasó todo el proceso de pruebas con resultados satisfactorios.


2. REPORTE DE APROBACIÓN

Reporte de Aprobación
Fecha de aprobación: 04/01/2017
Descripción: El Módulo de Planificación y Desarrollo Institucional a cargo del Ing. Jorge Malla cumplió con todas las fases de planificación propuestas en el Plan de PPQA, determinando que dicho modulo está apto para pasar a producción contando con la calidad óptima.



Elaborado

Lady Iñiguez



Revisado

Ing. Lissette López

6.4.3. Ejecución de pruebas

- **Pruebas Funcionales**

Los distintos casos de prueba se realizan en base a la descripción de Casos de Uso, es decir para cada Caso de Uso se realiza un Caso de prueba, para ello se ha hecho uso de la herramienta Testlink, en la cual se debe realizar los siguientes pasos para la ejecución de los casos de prueba.

PASO 1: Crear un Plan de Proyecto en la opción **(Gestión de Proyectos de Pruebas)**

PASO 2: Crear el Documento de Especificación de Requisitos en la opción **(Requisitos)** para poder enlazar con los casos de uso que más tarde servirán para crear los casos de prueba.

PASO 3: Se edita los casos de prueba en la opción **(Especificación de Pruebas)**, donde ingresaremos la acción a realizar y el resultado esperado.

PASO 4: Se crea un Plan de Pruebas en la opción **(Gestión de Planes de Prueba)**, para poder ejecutar los casos de prueba.

PASO 5: Se crea un build en la opción **(Gestión de Builds)**, para poder tener la versión del proyecto y permita ejecutar los casos de prueba.

PASO 6: Se ejecuta los casos de prueba en la opción **(Ejecución de Pruebas)**.

Luego de documentar todos los casos de prueba se observará un despliegue de los mismos en forma de árbol.

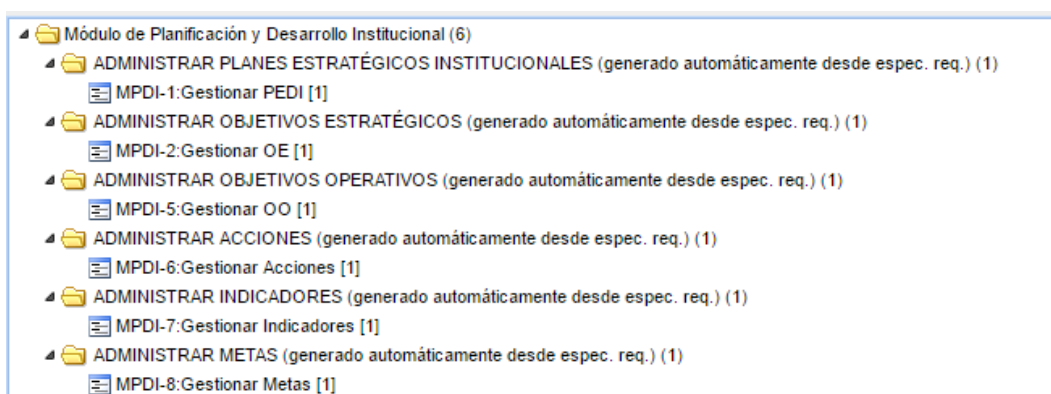


Figura 30. Árbol de Casos de Prueba

Para poder tener una comparativa del antes y después, se realizaron pruebas por parte de un pasante de la UTI sin la utilización de ninguna herramienta como generalmente

se lo maneja dentro del GDS, un claro ejemplo se refleja en la siguiente figura, donde en un solo caso de prueba se realiza la verificación de la mayoría de las funcionalidades del módulo, existiendo desorden y confusión al no saber a qué caso de uso o funcionalidad se refiere dicha observación, además de ser muy laborioso el hacer cada caso de prueba en tablas.

TÍTULO	Módulo Planificación		
PANTALLA	http://localhost:8087/planesEstrategicos		
OBJETIVOS	Verificar funcionalidades de la administración de los Planes Estratégicos Institucionales.		
RESULTADO(S) ESPERADO(S)			
VALIDADO:	SI	NO	PARCIALMENTE
OBSERVACIÓN(ES):			
<ol style="list-style-type: none"> 1. error : <code>at JSON.parse (<anonymous>) at CompuestoDetailComponent.getCatalogoItem (compuesto-detail.component.ts:311): Anexo 1</code> 2. Permite guardar Objetivos Operativos en blanco 3. No se elimina el Objetivo Estratégico 4. No se puede eliminar objetivos Operativos 5. Al pasar de Metas a Objetivos Estratégicos, y selecciona otro objetivo estratégico no se deshabilitan (Acciones, Indicadores, Metas) 6. Al pasar de otro botón o regresar a Planes Estratégicos y de seleccionar el plan se quedan habilitados los botones (Objetivos Operativos, Acciones, Indicadores, Metas) con sus ítems 7. Al eliminar (Objetivos Operativos, Acciones, Indicadores o Metas, desaparece el que le sigue, el eliminado se queda, para poder visualizar los cambios se debe cargar de nuevo la pantalla 8. Al crear un nuevo Objetivo Operativo permite guardar sin agregar un Nombre de Categoría o seleccionar una opción. 9. Permite guardar una acción sin los datos del Responsable: Anexo 2 10. Al tener agregado responsables en acciones, buscarlo con el botón de la barra de responsable, al no encontrar nada y guardarlo desaparecen los responsables ya agregados. 11. Al momento de la búsqueda al presionar la línea o el texto Buscar se presenta el error, pero si se realiza la búsqueda. Anexo 3 12. Permite seleccionar el botón eliminar sin habilitarse. Anexo 4 13. Al actualizar la pantalla se oculta la barra de navegabilidad. 14. Al eliminar un Plan los botones de editar y eliminar quedan activos, al presionarlo de nuevo se oculta el siguiente sin estar seleccionado para volver a visualizarlo se debe actualizar la página. 			
Después de las debidas correcciones de los errores antes mencionados la aplicación web está en correcto funcionamiento.			
NOMBRE TESTER:	Pasante: <u>Angel</u> Minga.		ID o RÚBRICA:

Figura 31. Ejemplo de realización de pruebas funcionales

Por su parte para este proyecto se realiza la primera ejecución de los casos de prueba con el uso de la herramienta Testlink, la misma que permite evidenciar de forma ordenada y entendible mediante la Matriz de Resultado que al igual que en la pruebas realizadas por el pasante y estas pruebas, la mayoría de los casos de prueba han resultado fallidos.

Número de casos de prueba → X=6 → 90%

Número de casos prueba exitosos → X=1

Número de casos prueba fallidos=5

$$X = \frac{1 \times 6}{90}$$

$$X = \frac{6}{90}$$

$$X = 0,06\%$$

Caso de Prueba	Prioridad	V 1.0	[Última Build] V 1.0	Última Ejecución
Suite de Pruebas: ADMINISTRAR ACCIONES (generado automáticamente desde espec. req.) (1 Item)				
MPDI-6: Gestionar Acciones [1]	Media	Fallado [v1]	Fallado [v1]	Fallado [v1]
Suite de Pruebas: ADMINISTRAR INDICADORES (generado automáticamente desde espec. req.) (1 Item)				
MPDI-7: Gestionar Indicadores [1]	Media	Fallado [v1]	Fallado [v1]	Fallado [v1]
Suite de Pruebas: ADMINISTRAR METAS (generado automáticamente desde espec. req.) (1 Item)				
MPDI-8: Gestionar Metas [1]	Media	Fallado [v1]	Fallado [v1]	Fallado [v1]
Suite de Pruebas: ADMINISTRAR OBJETIVOS ESTRATÉGICOS (generado automáticamente desde espec. req.) (1 Item)				
MPDI-2: Gestionar OE [1]	Media	Pasado [v1]	Pasado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR OBJETIVOS OPERATIVOS (generado automáticamente desde espec. req.) (1 Item)				
MPDI-5: Gestionar OO [1]	Media	Fallado [v1]	Fallado [v1]	Fallado [v1]
Suite de Pruebas: ADMINISTRAR PLANES ESTRATÉGICOS INSTITUCIONALES (generado automáticamente desde espec. req.) (1 Item)				
MPDI-1: Gestionar PEDI [1]	Media	Fallado [v1]	Fallado [v1]	Fallado [v1]

Figura 32. Matriz de resultado de pruebas v1.0

El 0,06% significa que básicamente **el Módulo de Planificación y Desarrollo Institucional tiene un 89,93% de fallas** y no está apto para pasar a producción. Para corregir tal porcentaje se hace uso de la plantilla de Informe de errores y se notifica al desarrollador para que resuelva las no conformidades encontradas.

La Figura 27 indica los nuevos resultados luego de realizar una 2da versión de las pruebas funcionales evidenciando que todos los casos de prueba pasaron.

Caso de Prueba	Prioridad	V 1.0	V 1.1	[Última Build] V 1.1	Última Ejecución
Suite de Pruebas: ADMINISTRAR ACCIONES (generado automáticamente desde espec. req.) (1 Item)					
MPDI-6: Gestionar Acciones [1]	Media	Fallado [v1]	Pasado [v1]	Pasado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR INDICADORES (generado automáticamente desde espec. req.) (1 Item)					
MPDI-7: Gestionar Indicadores [1]	Media	Fallado [v1]	Pasado [v1]	Pasado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR METAS (generado automáticamente desde espec. req.) (1 Item)					
MPDI-8: Gestionar Metas [1]	Media	Fallado [v1]	Pasado [v1]	Pasado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR OBJETIVOS ESTRATÉGICOS (generado automáticamente desde espec. req.) (1 Item)					
MPDI-2: Gestionar OE [1]	Media	Pasado [v1]	No Ejecutado [v1]	No Ejecutado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR OBJETIVOS OPERATIVOS (generado automáticamente desde espec. req.) (1 Item)					
MPDI-5: Gestionar OO [1]	Media	Fallado [v1]	Pasado [v1]	Pasado [v1]	Pasado [v1]
Suite de Pruebas: ADMINISTRAR PLANES ESTRATÉGICOS INSTITUCIONALES (generado automáticamente desde espec. req.) (1 Item)					
MPDI-1: Gestionar PEDI [1]	Media	Fallado [v1]	Pasado [v1]	Pasado [v1]	Pasado [v1]

Figura 33. Matriz de resultados de pruebas v1.1

Número de casos de prueba → $X=6 \rightarrow 90\%$

Número de casos prueba exitosos → $X=6$

Número de casos prueba fallidos = 0

De acuerdo a segunda versión de las pruebas realizadas se demuestra que **el módulo básicamente cumple todas las funcionalidades requeridas representadas en el 90% y está apto para pasar a las siguientes pruebas**, no se puede decir que el módulo cumplió un 100% debido a que un software no es 100% perfecto, por ello se toma como base de cumplimiento el 90%, dejando un margen de error del 10% puesto que dentro de cualquier etapa del ciclo de vida de un proyecto puede existir cambios.

En la Figura 28 se observa la documentación generada por la herramienta Testlink **MPDI_test_report-2016-12-27** en la primera fase de la prueba funcional (**v1.0**) y **MPDI_test_report-2017-01-04 (v1.1)** como resultado de la segunda fase de prueba de funcionalidad realizada, una vez corregidos todos los errores encontrados.

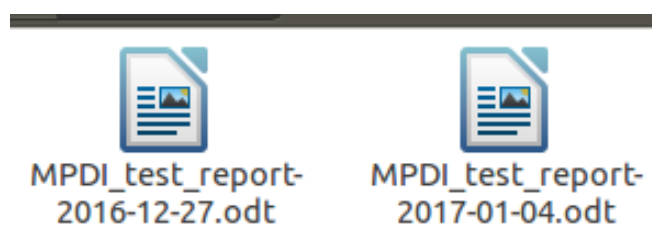


Figura 34. Documento de reporte de pruebas

Existen varios formatos para exportar los resultados en caso de querer informar directamente con los informes generados por Testlink como son: una página HTML, OpenOffice Write, Word, Excel, Graficos, etc.

- **Pruebas de Seguridad a nivel de aplicación**

Para las pruebas de seguridad a nivel de aplicación, se identificó el tipo de usuario el mismo que es “**Administrador de PDI**” y se validó que tenga acceso a las funcionalidades predefinidas para su rol.

TABLA XL. CASOS DE PRUEBA DE SEGURIDAD A NIVEL DE APLICACIÓN

REGISTRO DE CASOS DE PRUEBA				
Módulo de Planificación y Desarrollo Institucional				
Identificador de caso de rol	Versión	Tipo de prueba	Fecha de creación	Fecha de culminación
Administrador de PDI	v1.0	Seguridad a nivel de Aplicación	28/12/2016	28/12/2016
Funciones		Acciones autorizados		Resultado obtenido
Administrar PEDI		Crear PEDI		Exitoso
		Editar PEDI		Exitoso
		Buscar PEDI		Exitoso
		Eliminar PEDI		Exitoso
Administrar Objetivos Operativos		Crear Objetivos Operativos		Exitoso
		Editar Objetivos Operativos		Exitoso
		Buscar Objetivos Operativos		Exitoso
		Eliminar Objetivos Operativos		Exitoso
Administrar Acciones		Crear Acciones		Exitoso
		Editar Acciones		Exitoso
		Buscar Acciones		Exitoso
		Eliminar Acciones		Exitoso
Administrar Indicadores		Crear Acciones		Exitoso
		Editar Indicadores		Exitoso
		Buscar Indicadores		Exitoso
		Eliminar Indicadores		Exitoso
Administrar Metas		Crear Metas		Exitoso
		Editar Metas		Exitoso
		Buscar Metas		Exitoso
		Eliminar Metas		Exitoso

Como se puede observar se ha tenido un éxito en todas casos de prueba, concluyendo que no existen errores en el acceso a las funcionalidades predefinidas para ese rol.

- **Pruebas de Rendimiento**

Para las pruebas de rendimiento se pretende crear varios escenarios mediante la herramienta Jmeter, para determinar el tiempo de respuesta del sistema con un número de usuarios acorde a lo especificado en el documento ERS y probar también bajo escenarios no programados.

Escenario 1: Este caso de prueba debería simular el acceso de 100 usuarios simultáneos y el tiempo de respuesta para cada usuario debería ser de 1 segundo.

PASO1: Damos click derecho en Plan de pruebas->Add->Threads(Users)->**Thread Group**

Se configura el número de Usuarios que para este escenario son **100** y el periodo de tiempo de **1 segundo**.

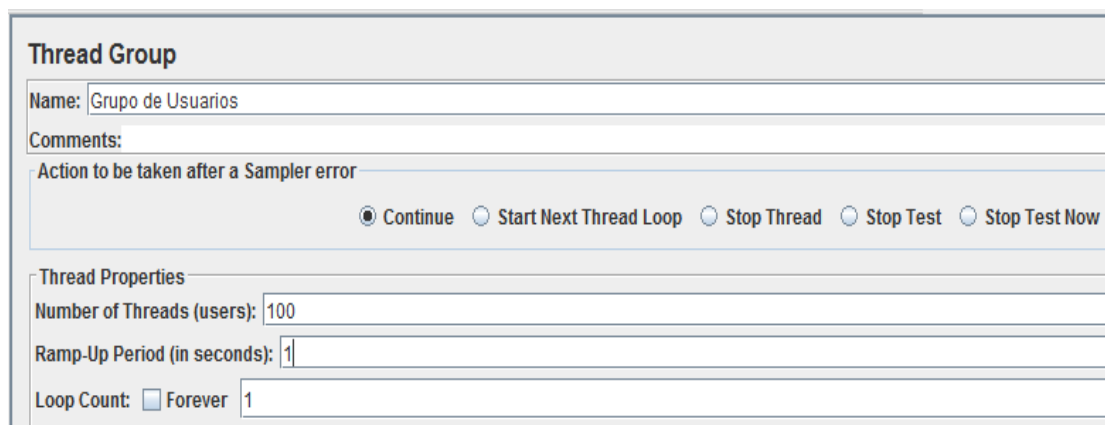


Figura 35. Thread Group

PASO2: Damos click derecho en Thread Group->Add->Sampler->**HTTP Request**

- Colocamos la dirección IP en Server Name or IP: **10.10.50.40**
- El puerto: **8000**
- Path: **/api/v1/planificacion/planes-estrategicos?page=1&numberItems=20**

No se trabaja específicamente con una petición Http/Https, puesto que aun el Módulo de Planificación y Desarrollo Institucional no está en producción, por tal razón se consume directo del api del mismo.

Figura 36. HTTP Request

PASO3: Se generan los reportes de la prueba en 4 tipos diferentes, el valor adicional que debe dar un Tester es en sí la interpretación de los resultados que arroja la ejecución del test.

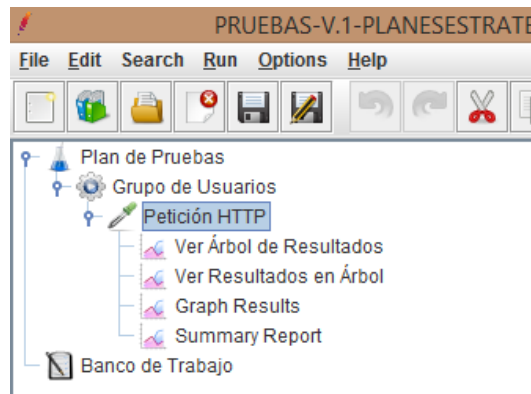


Figura 37. Reportes de resultados

En el caso específico de este ejemplo práctico los reportes definidos quedaron de la siguiente manera:

- **Árbol de Resultados:** La vista de árbol de resultados muestra un árbol de todas las respuestas del caso de prueba tanto las fallidas como las acertadas. Además de mostrar la respuesta se puede ver el tiempo que se tarda en llegar a obtener la respuesta.

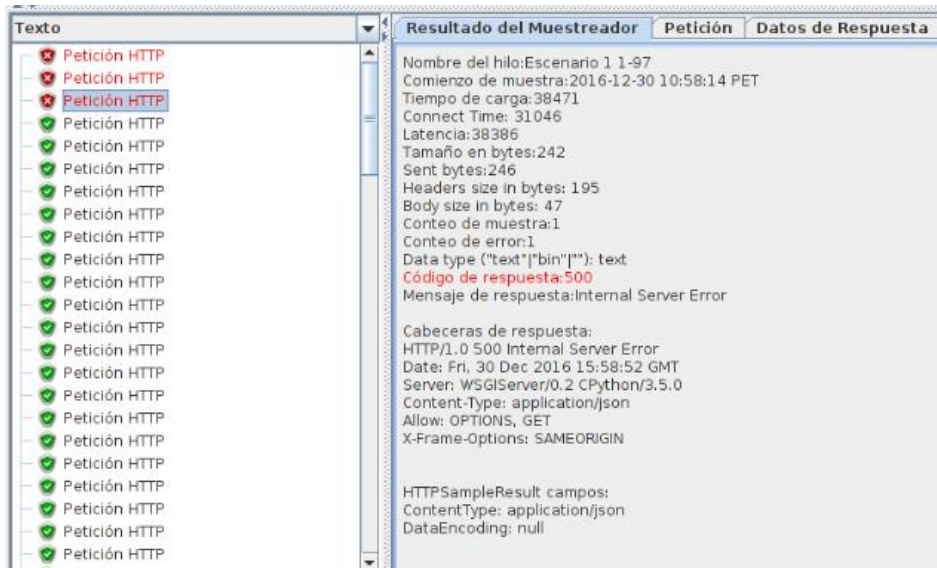


Figura 38. Árbol de resultados

- **Summary Report:** Crea una tabla para cada caso de prueba indicando los tiempos de respuesta, el tiempo mínimo, máximo, promedio, tasa de error y rendimiento aproximado.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...
Petición HTTP	100	75761	38139	87333	10191,02	3,00%	1,1/sec	44,99	0,27	40631,8
Total	100	75761	38139	87333	10191,02	3,00%	1,1/sec	44,99	0,27	40631,8

Figura 39. Summary Report

- **Tabla de resultados:** Crea una fila para cada resultado de la muestra al igual que el árbol de resultados además solo muestra las muestras principales medidas en milisegundos.

Muestra #	Tiempo de co...	Nombre del hilo	Etiqueta	Tiempo de M...	Estado	Bytes	Sent Bytes	Latency	Connect Time...
1	10:58:14.168	Escenario 1 ...	Petición HTTP	38139	✖	242	246	38134	31079
2	10:58:14.127	Escenario 1 ...	Petición HTTP	38354	✖	242	246	38326	31056
3	10:58:14.137	Escenario 1 ...	Petición HTTP	38471	✖	242	246	38386	31046
4	10:58:13.124	Escenario 1 ...	Petición HTTP	61016	✔	41881	246	60935	2
5	10:58:13.283	Escenario 1 ...	Petición HTTP	62425	✔	41881	246	62233	1
6	10:58:13.201	Escenario 1 ...	Petición HTTP	62906	✔	41881	246	62586	1
7	10:58:13.334	Escenario 1 ...	Petición HTTP	62784	✔	41881	246	62363	1
8	10:58:13.157	Escenario 1 ...	Petición HTTP	63139	✔	41881	246	62734	2
9	10:58:13.171	Escenario 1 ...	Petición HTTP	63483	✔	41881	246	63256	2
10	10:58:13.232	Escenario 1 ...	Petición HTTP	64100	✔	41881	246	63489	1
11	10:58:13.263	Escenario 1 ...	Petición HTTP	64196	✔	41881	246	63657	1
12	10:58:13.273	Escenario 1 ...	Petición HTTP	64432	✔	41881	246	64213	1
13	10:58:13.473	Escenario 1 ...	Petición HTTP	64288	✔	41881	246	63687	1
14	10:58:13.180	Escenario 1 ...	Petición HTTP	64643	✔	41881	246	64475	2
15	10:58:13.190	Escenario 1 ...	Petición HTTP	64815	✔	41881	246	64285	2
16	10:58:13.222	Escenario 1 ...	Petición HTTP	64796	✔	41881	246	64790	1
17	10:58:13.293	Escenario 1 ...	Petición HTTP	64997	✔	41881	246	64938	2
18	10:58:13.499	Escenario 1 ...	Petición HTTP	65294	✔	41881	246	64908	2
19	10:58:13.324	Escenario 1 ...	Petición HTTP	67872	✔	41881	246	66725	1
20	10:58:13.146	Escenario 1 ...	Petición HTTP	68433	✔	41881	246	68063	1
21	10:58:13.252	Escenario 1 ...	Petición HTTP	69049	✔	41881	246	68195	1
22	10:58:13.242	Escenario 1 ...	Petición HTTP	69062	✔	41881	246	68618	2
23	10:58:13.135	Escenario 1 ...	Petición HTTP	69794	✔	41881	246	69258	2
24	10:58:13.375	Escenario 1 ...	Petición HTTP	69659	✔	41881	246	69168	1
25	10:58:13.354	Escenario 1 ...	Petición HTTP	70126	✔	41881	246	70046	1
26	10:58:13.303	Escenario 1 ...	Petición HTTP	70260	✔	41881	246	69875	1
27	10:58:13.212	Escenario 1 ...	Petición HTTP	70422	✔	41881	246	70372	1
28	10:58:13.344	Escenario 1 ...	Petición HTTP	71087	✔	41881	246	70821	1
29	10:58:13.364	Escenario 1 ...	Petición HTTP	71210	✔	41881	246	71101	2
30	10:58:13.581	Escenario 1 ...	Petición HTTP	71432	✔	41881	246	71162	1

Scroll automatically?
 Child samples?
No. de Muestras 100
Última Muestra 87272
Media 75761
Desviación 10191

Figura 40. Tabla de resultados

Se han realizado 8 escenarios diferentes probando con cargas distintas, llegando a la conclusión que el escenario especificado como apto el mismo que está pintado con un color plateado llegando a ser el **Escenario 1** con 100 Usuarios en 1 segundo tuvo un margen de error del 3%, con una latencia máxima de 87 segundos en devolver una de las peticiones y con una latencia mínima de 38 segundos resultando no ser optima puesto que no es correcto demorarse 87 segundos en responder una petición, lo ideal esta entre 2-3 segundos.

Los escenarios que no obtuvieron ningún error fueron los **Escenario 3 (con un color verde), Escenario 6 y Escenario 7 (con un color naranja)**, pero los que menor latencia presentaron son **Escenario 6 y Escenario 7** con una **latencia mínima de 6,559 y 4,859** y una **latencia máxima en ambos casos de 9,961 y 9,983** respectivamente y un **rendimiento de 1.2**. Los escenarios restantes presentan demasiada latencia para ser considerados escenarios óptimos.

Además se observa un rendimiento estable ya sea en 100, 90 o 200 Usuarios soportados, con una latencia en el último caso de 200 usuarios de 3 segundos óptima para el usuario.

TABLA XLI. RESULTADO DE PRUEBAS DE RENDIMIENTO

Escenario	Módulo	Tipo de Prueba	Número de peticiones	Tiempo de envío de peticiones (segundos)	Latencia mínima (segundos)	Latencia máxima (segundos)	Número de peticiones incorrectas	Rendimiento (segundos)	Error	Resultado
Escenario 1	Módulo de Planificación y Desarrollo Institucional	Rendimiento	100	1	38,139	87,333	3	1.1	3%	Fallido
Escenario 2	Módulo de Planificación y Desarrollo Institucional	Carga	150 – en un bucle de 2 veces	1	0,643	122,591	106	1.4	35.33%	Fallido
Escenario 3	Módulo de Planificación y Desarrollo Institucional	Rendimiento	90	2	61,126	78,575	Ninguna	1.1	0%	Exitoso
Escenario 4	Módulo de Planificación y Desarrollo Institucional	Rendimiento	200	1	35,600	127,342	23	1.6	47.50%	Fallido
Escenario 5	Módulo de Planificación y Desarrollo Institucional	Rendimiento	185	1	37,708	127,342	17	1.4	32.97%	Fallido
Escenario 6	Módulo de Planificación y Desarrollo Institucional	Carga	10 - en un bucle de 10 veces	1	6,559	9,961	Ninguna	1.2	0%	Exitoso
Escenario 7	Módulo de Planificación y Desarrollo Institucional	Carga	10 - en un bucle de 20 veces	2	4,859	9,983	Ninguna	1.2	0%	Exitoso
Escenario 8	Módulo de Planificación y Desarrollo Institucional	Carga	100 - en un bucle de 2 veces	3	0,771	102,447	6	1.1	3%	Fallido

Cabe recalcar que las pruebas de rendimiento no se realizaron desde un servidor puesto que el modulo aún no está en producción como hubiera sido lo ideal, sino directo desde la dirección IP, determinando que es uno de los aspectos para evidenciar una latencia muy alta en todos los casos de pruebas. Recomendando que en caso de volver a realizar estas pruebas se tome en cuenta la optimización de líneas de código para evitar código basura que consuma más de lo debido, otra de las apreciaciones es evitar acceder a la base de datos a menos que realmente sea necesario evitando cuellos de botella cuando existe mucha carga y por último se recomienda realizar una réplica del servidor de base de datos permitiendo de esta manera que los usuarios puedan acceder más rápido a la información disminuyendo el tiempo de latencia y mejorando el rendimiento de las consultas.

6.4.4. Análisis del mejoramiento al implementar CMMI-DEV

Para realizar el análisis y comparativa de resultados se toma en cuenta todas las actividades realizadas, desde el inicio de la propuesta hasta la validación de la misma haciendo hincapié en el módulo de Planificación y Desarrollo Institucional.

TABLA XLII. COMPARATIVA DE LAS ACTIVIDADES REALIZADAS CON EL PROCESO ACTUAL vs LA PROPUESTA

Desarrollo del Módulo “Planificación y Desarrollo Institucional”		
PROCESO ACTUAL	vs	PROCESO PROPUESTO
2 semanas fase de análisis 2 semanas fase de diseño 8 meses fase de desarrollo 1 semana pruebas <ul style="list-style-type: none"> Pruebas funcionales $T_{\text{módulo}} = 2\text{sem} + 2\text{sem} + 8\text{ms} + 1\text{sem}$ $T_{\text{módulo}} = 9 \text{ mes y } 1 \text{ sem}$		2 semanas fase de análisis <ul style="list-style-type: none"> Validación del ERS Validación de Prototipo 2 semanas fase de diseño <ul style="list-style-type: none"> Validación de CU 8 meses fase de desarrollo 4 días pruebas <ul style="list-style-type: none"> Funcionales Seguridad a nivel de aplicación Rendimiento $T_{\text{módulo}} = 2\text{sem} + 2\text{sem} + 8\text{ms} + 4\text{días}$ $T_{\text{módulo}} = 9 \text{ mes y } 4 \text{ días}$
		Notas: <ul style="list-style-type: none"> El tiempo presentado es una valoración propia y estimada de acuerdo a las fecha propuestas en las plantillas evaluadas y de la entrevista realizada al Ing. encargado de dicho módulo. No se puede evidenciar la diferencia de tiempo que puede existir al implementar o no CMMI puesto que el módulo propuesto como piloto de pruebas ya estaba en sus últimas fases de desarrollo. Se puede evidenciar el aseguramiento de la calidad realizado en cada fase de desarrollo del mismo.

TABLA XLIII. RESOLUCIÓN DE PROBLEMAS ENCONTRADOS EN EL GDS

PROBLEMAS	PRIORIDAD	DECISIÓN TOMADA	MEJORAS	CUMPLE/NIVEL	
				Total	Medio
Escasa documentación formalizada de procesos y estándares.	3	Levantamiento de plantillas.	Se documentó todo el proceso de desarrollo de software, quedando como fuente de información para futuros proyectos.	✓	
Inexistencia de un plan PPQA.	3	Propuesta de un Plan de PPQA.	Garantizó el cumplimiento de las fases para asegurar la calidad del producto.	✓	
Desconocimiento del personal en metodologías y modelos de mejora de procesos.	3	Dar a conocer al personal las diversas metodologías para la mejora de procesos y acerca del modelo CMMI.	El personal puede decidir en implementar un modelo e iniciar el camino hacia una mejora de procesos optando por una certificación.		✓
Personal no calificado en herramientas tecnológicas para el aseguramiento de la calidad.	3	Dar a conocer al personal los distintos tipos de herramientas para asegurar la calidad del software.	Se proporcionó documentación del resultado obtenido mediante el uso de las 2 herramientas, evidenciando varios aspectos que se estaban pasando por alto, los mismos que fueron corregidos.		✓

TABLA XLIV. COMPARATIVA DE LO QUE SE TIENE Y LO QUE SE PROPUSO

Estado actual	Decisión tomada	Implicación	Meta cumplida de CMMI-DEV
Carencia de políticas de Aseguramiento de la Calidad.	Definición de políticas de Aseguramiento de la Calidad. – Utilización de la norma del Comité Internacional de Cualificaciones de Pruebas de Software (ISTQB)	Formalización de las políticas.	SG 1. Evaluar objetivamente los procesos y los productos de trabajo. (En contraste a los procesos, estándares y procedimientos definidos).
No consta con todos los roles para cubrir cada fase del ciclo de vida de un proyecto.	Determinar roles y responsabilidades. – Utilización de la norma del Comité Internacional de Cualificaciones de	Formalización de roles y responsabilidades.	– SP 1.1 Evaluar objetivamente los procesos.

	Pruebas de Software (ISTQB)		<ul style="list-style-type: none"> – SP 1.2 Evaluar objetivamente los productos de trabajo.
No existen procesos definidos.	Establecimiento de nuevos procesos con los involucrados de la UTI. <ul style="list-style-type: none"> – Utilización del estándar Modelo y Notación de procesos de Negocio (BMPN) 	Cambio de procesos.	
Documentación nula en el proceso de desarrollo de software.	Levantamientos de plantillas. <ul style="list-style-type: none"> – Plan de PPQA ref. estándar IEEE 730-2002. – Checklist de validación de requisitos, ref. estándar IEEE 830-1998. – Checklist de validación de casos de uso, ref. estándar UML. – Plan de pruebas e Informe de errores ref. ISO/IEC/IEEE 29119-3:2013. 	Hacer uso de las plantillas propuestas.	SG 2. Proporcionar una visión objetiva. (Informar las no conformidades y establecer registros). <ul style="list-style-type: none"> – SP 2.1 Comunicar y resolver las no conformidades. – SP 2.2 Establecer los registros.
Ciclo de pruebas poco fiable.	Propuesta de herramientas para asegurar la calidad. <ul style="list-style-type: none"> – Uso de herramientas Testlink y Jmeter 	Hacer uso de herramientas que ayuden al aseguramiento de la calidad.	

De acuerdo a lo expuesto en las tablas anteriores se puede evidenciar que haber aplicado CMMI-DEV específicamente el área de Aseguramiento de la Calidad del Proceso y del Producto en el “Módulo de Planificación y Desarrollo Institucional” permitió contrastar problemas existentes y mejorarlos, como la falta de realización de pruebas, uso de herramientas para asegurar la calidad, no tener documentación durante el proceso de desarrollo, etc. Al haber mejorado todos estos inconvenientes se determina que CMMI-DEV proporcione varios aspectos que en caso que la UTI deseará implementar el mismo podría ser una base o una iniciativa para mejorar todo el procesos de desarrollo de software.

7. Discusión

7.1. Desarrollo de la propuesta

El presente trabajo de titulación ha sido desarrollado con el fin de proponer una mejora de procesos en Área de Aseguramiento de la Calidad de Software para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja. Para poder cumplir con el tema propuesta se cumplieron a cabalidad cada uno de los objetivos planteados a continuación:

7.1.1. Realizar un análisis de la situación actual del departamento de desarrollo de software de la Unidad de Telecomunicaciones e Información con respecto al aseguramiento de la calidad del software.

En esta fase la misma que de acuerdo al modelo IDEAL es “Diagnóstico” se realizó un análisis de la situación actual del departamento de desarrollo de software en base al Aseguramiento de la Calidad de Software, mediante técnicas como análisis FODA, árbol causa efecto, árbol de objetivos y entrevistas para mayor información (**Anexo 2**); así también sobre políticas de Aseguramiento de la Calidad de Software, procesos actuales de desarrollo de software y determinar el problema central que afecta al departamento de desarrollo de software de la UTI, priorizando los problemas más importantes específicamente los que forman parte directa del proyecto en ejecución.

7.1.2. Desarrollo de la propuesta de políticas, estándares, procesos y procedimientos de Aseguramiento de la Calidad de Software que se adecue a las necesidades del departamento de software de la UTI.

En esta fase la misma que de acuerdo al modelo IDEAL es “Establecer” se definieron roles y responsabilidades de los miembros que conformarán el grupo de PPQA, tomando como base la información proporcionada por el Comité Internacional de Cualificaciones de Pruebas de Software (ISTQB) y otros trabajos relacionados con lo mismo, también se planteó políticas para el área PPQA que se deberían cumplir dentro del GDS de la UTI, seguido se realizó el mejoramiento de procesos teniendo como base los procesos actuales (**Anexo 4**) para los cuales se hizo uso del estándar Modelo y Notación de Procesos de Negocio (BPMN) los que fueron aprobados por el Subdirector de Desarrollo de Software (**Anexo 6**). Posteriormente se realizó el levantamiento de plantillas las mismas que fueron aprobadas por la Analista/Desarrolladora Ing. Lissette López (**Anexo 7**) para el levantamiento de las mismas se hizo uso de los estándares

como: IEEE 730-202 para la plantilla plan de aseguramiento de la calidad del proceso y del producto, IEEE-830-1998 para la plantilla checklist de validación de requisitos , UML para la plantilla checklist de validación de casos de uso e ISO/IEC/IEEE 29119-3:2013 (**Anexo 8**) para la plantilla plan de pruebas y la plantilla informe de errores.

7.1.3. Analizar herramientas a utilizar para el proceso de aseguramiento de la calidad de software y recomendar las adecuadas.

El análisis de herramientas tanto Comerciales como Open Source para pruebas Funcionales, Seguridad y Rendimiento se realizó seleccionando las herramientas más conocidas y utilizadas. Para las pruebas Funcionales se seleccionó 8 herramientas: Testlink, Selenium, Test Complete, Rational Functional Tester, Microsoft Test Manager, Unified Functional Tester, WET y SoapUI, la comparación se realizó en base a 10 lineamientos como si es Open Source o Comercial, Plataforma de funcionamiento, Soporte, Manuales de usuario e instalación, entre otros, al final se concluyó que las mejores herramientas debido al alto grado de soporte y gestión de pruebas son Testlink, y Test Complete, entre ellas la que se destaca y se acoplan a las pruebas Funcionales de caja negra (**Anexo 9**) es la herramienta Testlink. Para las pruebas de Seguridad se seleccionó 7 herramientas: OWASP ZAP, Burp Suite, Snort, WireShark, Solarwinds, Suricats y GFI LANguard, la comparación se realizó en base a 9 lineamientos como si es Open Source o Comercial, Plataforma de funcionamiento, Soporte, entre otros, llegando a la conclusión que aunque entre estas herramientas ninguna se acopla a las pruebas de seguridad (**Anexo 9**) las mejores herramientas debido al número de clientes entre empresas grandes y pequeñas, manuales de instalación y soporte son OWASP ZAP y Snort. Para las pruebas de Rendimiento se seleccionó 6 herramientas: FunkLoad, NeoLoad, Jmeter, Gatting, Qtest5.0, MultiMechanize, la comparación se realizó en base a 11 lineamientos como que Protocolos soporta, Plataforma de funcionamiento, Visualización en tiempo real, entre otros, llegando a la conclusión que las mejores herramientas son NeoLoad y Jmeter entre ellas se destaca Jmeter la misma que se acopla al módulo a probar.

7.1.4. Validar la propuesta con un proyecto en ejecución en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.

Se ejecutó el piloto de pruebas haciendo uso de cada uno de los procesos y actividades, fue asignado por parte del Director de la Unidad de telecomunicaciones e Información

Ing. Milton Labanda y el Subdirector de Desarrollo de Software Edison Coronel el sistema en desarrollo Sistema de Información Académico Administrativo Financiero (SIAAF) de la Universidad Nacional de Loja, específicamente el Módulo de Planificación y Desarrollo Institucional. Se siguió cada uno de los procesos planteados, realizando la validación de las plantillas las mismas que fueron revisadas y validadas por la Analista/Desarrolladora Ing. Lissette López, también se realizó la instalación de las herramientas antes seleccionadas de acuerdo al tipo de prueba a realizar, Testlink se utilizó para las pruebas funcionales tomando como referencia la descripción de Casos de Uso para plantearse los pasos a seguir, mientras que Jmeter para las pruebas de rendimiento realizándolas no desde el servidor sino directamente desde la dirección IP montando 8 escenarios distintos con cargas distintas, las pruebas de seguridad se las realizaron manualmente debido a no encontrar una herramienta que se acople a las necesidades de la prueba. Los resultados de las pruebas Funcionales permitieron descubrir que ciertas acción dentro de un caso de prueba no se cumplía por lo cual se determinó que era un caso de prueba fallado, notificando al Desarrollador los errores encontrados siguiendo paso a paso los procesos ya establecidos, por su parte la prueba de seguridad todos los casos de prueba fueron exitosos, y por ultimo las pruebas de rendimiento se determinó con la ejecución de los 8 escenarios que en la mayoría de los casos se observó una latencia muy alta recomendando optimizar líneas de código y evitar acceder a la base de datos a menos que realmente sea necesario.

7.2. Valoración técnica económica ambiental

El presente trabajo de titulación denominado “Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI V1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.”, se lo considera factible en los tres aspectos técnico, económico y ambiental puesto que en el ámbito tecnológico el desarrollo del mismo no implica el uso de dispositivos o equipos costosos, simplemente se necesita de un ordenador. En el ámbito económico el proyecto es viable debido a que el desarrollo del mismo no presenta ningún gasto excesivo puesto que las herramientas y estándares utilizados son libres y gratuitos. Y finalizando en el ámbito ambiental el trabajo de titulación es un trabajo de titulación de carácter formativo por ello no causa daños ambientales.

8. Conclusiones

- Al haber aplicado Aseguramiento de la Calidad del Proceso y del Producto en el plan piloto permitió evidenciar una mejora notable del producto esto debido a que desde sus inicios se cumplió con los procesos de PPQA antes establecido.
- El modelo IDEAL es el que mejor se adapta para integrarlo con CMMI (Capability Maturity Model Integration), esto debido a que desde su concepción fue diseñado para aplicarlo con el Modelo de Mejora de Procesos.
- Emplear técnicas como el análisis FODA, árbol causa-efecto, árbol de objetivos, son importantes para poder identificar problemas y saber qué acciones tomar para solucionar los mismos.
- La utilización del estándar BPMN facilita el diseño de procesos de desarrollo de software de una manera unificada y estandarizada los mismos que permiten realizar Aseguramiento de la Calidad en todo el ciclo de vida de un producto.
- Utilizar herramientas como Jmeter y Testlink para pruebas de Rendimiento y Funcionales, permite optimizar gran parte de las pruebas, evitando la realización manual de las mismas obteniendo resultados tempranos y eficaces.
- El modelo CMMI e IDEAL son los modelos que mejor se adaptan para conseguir una mejora de procesos en el área de Aseguramiento de la Calidad del Proceso y del Producto.

9. Recomendaciones

- Al desarrollar un proyecto de software se debe tener conversaciones constantes con todos los involucrados del GDS en cada etapa del proyecto, debido a que es el primer paso para poder conseguir una mejora dentro de una organización.
- Al empezar una mejora de procesos de software es oportuno determinar al inicio un Grupo de SPI (Software Process Improvement) quienes serán los encargados de cumplir con todas las fases de un proyecto que aseguren la Calidad del producto.
- El GDS de la UTI antes de iniciar un proyecto de software debe tener establecida una línea base la que se irá actualizando en los próximos proyectos y permitirá renovarla de acuerdo a las mejoras establecidas.
- El GDS debe contar con una estructura orgánica que incorpore un Líder de Proyecto, Líder de PPQA, Testers, necesarios para la ejecución de un proyecto software, los mismos que velaran por la calidad del producto.
- Utilizar las plantillas definidas en este proyecto puesto que permiten evidenciar falencias y corregirlas antes de que el producto pase a producción garantizando calidad, las mismas podrán ser mejoradas para proyectos futuros.
- Implementar la propuesta de SPI realizada en el área de PPQA, para que de esta manera el GDS pueda incorporar procesos que aseguren la Calidad del producto.
- Empezar una certificación del nivel 2 de CMMI-DEV v1.3 para garantizar la calidad de los productos de software.

10. Bibliografía

- [1] M. A. Mata Muñoz, «Análisis de estándares y modelos de referencia de mejores prácticas,» Madrid, 2007.
- [2] S. E. Institute, «CMMI para Desarrollo,» Noviembre 2010. [En línea]. Available: <http://www.sei.cmu.edu/library/assets/whitepapers/Spanish%20Technical%20Report%20CMMI%20V%201%203.pdf>. [Último acceso: 26 04 2016].
- [3] ISO, «Norma Internacional ISO 9001,» [En línea]. Available: http://www.fbioyf.unr.edu.ar/evirtual/pluginfile.php/106579/mod_resource/content/0/ISO%209001-2000%20%28ESP%29.pdf. [Último acceso: 20 05 2017].
- [4] M. Villa , M. Ruiz y I. Ramos, 2004. [En línea]. Available: http://rabida.uhu.es/dspace/bitstream/handle/10272/8773/Modelos_de_evaluacion.pdf?sequence=2. [Último acceso: 20 05 2017].
- [5] A. Alarcón, J. González y J. Rodríguez, «Guía para pymes desarrolladoras de software, basada en la norma ISO/IEC 15504,» *Revista Virtual Universidad Católica del Norte*, vol. 34, pp. 285-313, 2011.
- [6] T. J. F. H. Roberto José Herrera Acosta, *Seis Sigma Métodos Estadísticos y sus Aplicaciones*, 2011.
- [7] G. Rodríguez Villanueva, «Informe de Actividades 2010-2011,» México, 2011.
- [8] F. Ramos , O. Torres , N. Sánchez y M. Alba, « Implantación de CMMI nivel de madurez 2 en una,» *Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 6, nº 3, pp. 35-46, 2010.
- [9] P. Sistema, «Panel Sistema inicia la renovación de la acreditación en CMMI, 2017,» 2017. [En línea]. Available: <http://www.panel.es/metodologia/panel-sistemas-inicia-el-proceso-de-renovacion-de-la-acreditacion-en-cmmi/>. [Último acceso: 20 05 2017].
- [10] C. I. P. A. Results. [En línea]. Available: <https://sas.cmmiinstitute.com/pars/pars.aspx>. [Último acceso: 22 05 2016].
- [11] IZERTIS, «IZERTIS,» [En línea]. Available: <http://www.izertis.com/actualidad/535-izertis-obtiene-la-certificacion-iso-iec-15504>. [Último acceso: 20 05 2017].
- [12] Bitware, «Servicios de innovación,» [En línea]. Available: <https://www.bitware.es/Catalogo/Catalogo.pdf>. [Último acceso: 20 05 2017].
- [13] M. PeopleNet, «Grupo Bimbo,» 2012. [En línea]. Available: http://www.meta4.es/files/case-studies/bimbo_2012_esp.pdf. [Último acceso: 20 05 2017].
- [14] M. L. Pereyra, «BSI Group,» [En línea]. Available: <https://www.bsigroup.com/LocalFiles/es-MX/casoExitoV2.pdf>. [Último acceso: 20 05 2017].
- [15] A. Group, «BSI Group,» [En línea]. Available: <https://www.bsigroup.com/Documents/iso-14001/case-studies/BSI-ISO-14001-ISO-9001-case-study-APS-UK-EN.pdf?epslanguage=es-MX>. [Último acceso: 20 05 2017].
- [16] I. Ecuador, «Empresas con ISO-9001,» [En línea]. Available: http://www.normalizacion.gob.ec/wp-content/uploads/downloads/2013/11/listado_empresas_2013.pdf. [Último acceso: 22 05 2017].
- [17] B. Group, «BSI Group,» [En línea]. Available: <https://www.bsigroup.com/Documents/iso-9001/case-studies/BSI-ISO-9001->

- case-study-Maurice-Phillips-Contracts-UK-EN.pdf?epslanguage=es-MX. [Último acceso: 20 05 2017].
- [18] E. Oérez y M. García, «Implementación de la metodología DMAIC-Seis Sigma en el envasado de licores en Fanal,» *SciELO*, vol. 27, nº 3, pp. 88-106, 2014.
- [19] S. Sivakumar y K. Muthusamy, «Critical Success Factors in Six Sigma Implementation - A Case Study of MNCs in Malaysia,» *IEEE Xplore*, pp. 536-540, 2011.
- [20] M. Llaneza, G. Dapozo, C. Greiner y M. Estayno, «Análisis comparativo de modelos de calidad orientado al desarrollo de software,» *Redalyc*, pp. 601-605, 2013.
- [21] G. Solarte, L. Muñoz y B. Arias, «Modelos de Calidad para procesos de software,» *Redalyc*, vol. XV, nº 42, pp. 375-379, 2009.
- [22] I. B. Romero, «Seis Sigma: una herramienta estratégica para la calidad,» de 2º *Congreso Latinoamericano*, Bariloche, 2004.
- [23] S. I. Durán Pérez, 2014. [En línea]. Available: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/5840/TESIS.pdf?sequence=1>. [Último acceso: 21 05 2017].
- [24] J. F. E. Hernández, «Biblioteca Central Universidad de San Carlos de Guatemala,» Febrero 2015. [En línea]. Available: http://biblioteca.usac.edu.gt/tesis/08/08_1484_Q.pdf. [Último acceso: 08 06 2016].
- [25] J. C. Carro y D. Nápoles, «Modelos de Calidad para Desarrollo de Software, Comparación,» Habana, Cuba, 2007.
- [26] C. I. García Romero, «catarina.udlap.mx,» [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/garcia_r_ci/. [Último acceso: 20 05 2017].
- [27] G. García Mireles y F. García Rubio, «Armonización de Modelos de Calidad del Software,» *EPISTEMUS*, pp. 41-46, 2013.
- [28] U. S. Tomas, 26 7 2012. [En línea]. Available: http://soda.ustadistancia.edu.co/enlinea/gonzalogutierrez_gestiondeproducciony calidad/estndares_de_calidad.html. [Último acceso: 5 4 2016].
- [29] Q. C. R. L. Garcia Manuel, «Mejora Continua de la Calidad en los Procesos,» *redalyc.org*, vol. vol. 6, nº núm. 1, pp. 89-94, 2003.
- [30] T. R. Garraza, «PDCA,» Pamplona, 2016.
- [31] S. B. Rodrigo Gómez Montoya, «Seis Sigma: Un enfoque teórico y aplicado en el ámbito empresarial basándose en información científica,» 2011. [En línea]. Available: <http://repository.lasallista.edu.co/dspace/bitstream/10567/515/1/13.%20223-242.pdf>. [Último acceso: 27 04 2016].
- [32] B. E. M. Ortiz, «DEFINICIÓN DE PLANES DE ACCIÓN COMO RESULTADO DE LAS EVALUACIONES DE PROCESO SOFTWARE EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS SOFTWARE,» Enero 2009. [En línea]. Available: http://jupiter.utm.mx/~tesis_dig/10722.pdf. [Último acceso: 30 04 2016].
- [33] S. E. Institute, Febrero 1996. [En línea]. Available: <http://www.sei.cmu.edu/reports/96hb001.pdf>. [Último acceso: 02 01 2016].
- [34] K. E. T. Arias, 2007. [En línea]. Available: <http://bibdigital.epn.edu.ec/bitstream/15000/586/1/CD-0979.pdf>. [Último acceso: 04 01 2016].

- [35] Bizagi, «BPMN,» 2014. [En línea]. Available: <http://resources.bizagi.com/docs/BPMNbyExampleSPA.pdf>. [Último acceso: 20 08 2016].
- [36] S. I. d. G. Vasco, 09 10 2013. [En línea]. Available: http://www.ejie.eus/contenidos/informacion/herramientas_ejie/es_0213/adjuntos/OTC-MAU%20v_1.0%20Testlink.pdf. [Último acceso: 25 11 2016].
- [37] SeleniumHQ. [En línea]. Available: <http://www.seleniumhq.org/>. [Último acceso: 11 06 2016].
- [38] T. Complete, «SMARTBEAR,» [En línea]. Available: <https://smartbear.com/product/testcomplete/features/>. [Último acceso: 25 11 2016].
- [39] IBM. [En línea]. Available: ftp://public.dhe.ibm.com/software/rationalsdp/v7/rft/7012/docs/install_instruction/pdf/RFT_install_es.pdf. [Último acceso: 01 11 2015].
- [40] Globe. [En línea]. Available: <https://www.globetesting.com/2012/08/microsoft-test-manager-y-la-calidad-del-software/>. [Último acceso: 03 11 2016].
- [41] H. U. F. Tester. [En línea]. Available: <https://www.globetesting.com/2012/03/comparativa-de-herramientas-para-pruebas-automaticas/>. [Último acceso: 05 11 2016].
- [42] Wet. [En línea]. Available: <http://wet.qantom.org/home.html>. [Último acceso: 03 11 2016].
- [43] SoapUI. [En línea]. Available: <https://www.soapui.org/open-source.html>. [Último acceso: 02 11 2016].
- [44] O. ZAP. [En línea]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project#tab=Get_Involved. [Último acceso: 04 11 2016].
- [45] B. Suite. [En línea]. Available: <https://portswigger.net/burp/>. [Último acceso: 01 11 2016].
- [46] Snort. [En línea]. Available: <https://www.snort.org/>. [Último acceso: 03 11 2016].
- [47] WireShark. [En línea]. Available: <https://www.wireshark.org/#learnWS>. [Último acceso: 05 11 2016].
- [48] Solarwinds. [En línea]. Available: http://web.swcdn.net/creative/pdf/datasheets/ES/1201_sw_npm_datasheet_web_mx.pdf. [Último acceso: 03 11 2016].
- [49] Solarwinds. [En línea]. Available: <http://www.solarwinds.com/es/>. [Último acceso: 04 11 2016].
- [50] N. Security. [En línea]. Available: https://www.gfi.com/~/media/files/datasheets/lanss/gfilanguardbrochure4_es_gen.ashx. [Último acceso: 04 11 2016].
- [51] Apache. [En línea]. Available: <http://jmeter.apache.org/>. [Último acceso: 02 11 2016].
- [52] FunkLoad. [En línea]. Available: <https://pypi.python.org/pypi/funkload/>. [Último acceso: 01 11 2016].
- [53] NeoLoad. [En línea]. Available: <http://www.neotys.com/neoload/overview>. [Último acceso: 01 11 2016].
- [54] Gatling. [En línea]. Available: <http://gatling.io/#/why>. [Último acceso: 01 11 2016].
- [55] Q. security. [En línea]. Available: http://www.quotium.com/qttest/qttest_benefits/. [Último acceso: 02 11 2016].

- [56] M. Mechanize. [En línea]. Available: <https://multi-mechanize.readthedocs.io/en/latest/#performance-load-tests-in-python>. [Último acceso: 03 11 2016].
- [57] ISO. [En línea]. Available: <http://www.normas9000.com/iso-9000-59.html>. [Último acceso: 13 11 2016].
- [58] J. E. A. Morales, 2010. [En línea]. Available: http://www.conductitlan.net/psicologia_organizacional/la_mejora_continua.pdf. [Último acceso: 05 11 2016].
- [59] S. G. y. S. E. d. Á. d. A. y. R. d. A. d. Valladolid, 09 2003. [En línea]. Available: <http://www.aciamericas.coop/IMG/mejoracontinua.pdf>. [Último acceso: 05 11 2016].
- [60] A. C. K. S. Aldo Dagnino, «Metamorfosis Adaptación rápida a los cambios empleados en el modelo de mejora IDEAL,» *ABB*, p. 6, 2009.
- [61] H. W. K. T. Karlheinz Kautz, Enero 2009. [En línea]. Available: <http://homepages.dcc.ufmg.br/~rodolfo/es-1-06/umCasolIDEALparaPequenaEmpresa.pdf>. [Último acceso: 08 06 2016].
- [62] S. V. Jeremy Weinstein, «The PDCA Continuous Improvement Cycle Module 6.4,» summer 2004. [En línea]. Available: http://ocw.mit.edu/courses/engineering-systems-division/esd-60-lean-six-sigma-processes-summer-2004/lecture-notes/6_3_pdca.pdf. [Último acceso: 07 06 2016].
- [63] «How To Roll the Deming Wheel,» 15 7 2009. [En línea]. Available: <http://www.itsmsolutions.com/newsletters/DITYvol5iss28.pdf>. [Último acceso: 07 06 2016].
- [64] J. C. P. José Costas, «Entender el ciclo PDCA de mejora continua,» *Asociación Española para la Calidad (QAEC)*, p. 4, 2010.
- [65] H. J. L. Juiña, «Repositorio Digital EPN,» 2009. [En línea]. Available: <http://bibdigital.epn.edu.ec/bitstream/15000/1467/1/CD-2262.pdf>. [Último acceso: 08 06 2016].
- [66] L. F. Scalone, Junio 2006. [En línea]. Available: <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>. [Último acceso: 08 06 2016].
- [67] D. P. K. K. P. M. Sokovic, «Quality Improvement Methodologies-PDCA Cycle, RADAR Matriz, DMAIC and DFSS,» *Journal*, vol. 43, 2010.
- [68] «Universidad de las Ciencias Informáticas,» 2012. [En línea]. Available: <http://www.uci.cu/?q=entorno-productivo>. [Último acceso: 04 05 2016].
- [69] DOAJ, «Experiencias del programa de mejora de procesos en la Universidad de las Ciencias Informáticas,» 2012. [En línea]. Available: <https://doaj.org/article/1476aca061a140d0820c57055bad35fd>. [Último acceso: 04 05 2016].
- [70] «Softtek,» [En línea]. Available: <http://www.softtek.com/es/approach/modelo-de-calidad>. [Último acceso: 04 05 2016].
- [71] «TATA CONSULTANCY SERVICES,» [En línea]. Available: http://www.tcs.com/about/heritage_values/Pages/default.aspx. [Último acceso: 04 05 2016].
- [72] Novutek. [En línea]. Available: <http://www.novutek.com/cmml/>. [Último acceso: 22 05 2016].
- [73] E. Aldunate, «Arboles Diagnostico,» 29 05 2008. [En línea]. Available: http://www.cepal.org/ilpes/noticias/noticias/9/33159/Arboles_Diagnostico.pdf. [Último acceso: 10 07 2016].

- [74] I. S. T. Q. Board, 30 03 2010. [En línea]. Available: http://www.sstqb.es/ficheros/sstqb_file95-a69acf.pdf. [Último acceso: 18 11 2016].
- [75] V. M. R. y. L. S. Gaona, 10 09 2010. [En línea]. Available: <http://dspace.utpl.edu.ec/bitstream/123456789/1590/3/Mogrovejo%20Viviana%3b%20Sarango%20Luc%C3%ADa.pdf>.
- [76] I. o. E. a. E. E. (IEEE), 23 09 2002. [En línea]. Available: <http://scihub.bz/10.1109/IEEESTD.2002.94130>. [Último acceso: 29 11 2016].
- [77] I. o. E. a. E. Engineers, 20 10 1998. [En línea]. Available: <http://scihub.bz/10.1109/IEEESTD.1998.88286>. [Último acceso: 29 11 2016].
- [78] A. G. Pere Ponsa, «Diseño de pantalla,» [En línea]. Available: <http://www.epsevg.upc.edu/hcd/material/lecturas/pantalla.pdf>. [Último acceso: 2016 11 29].
- [79] G. d. Chile, «Diseño Web y Estándares,» [En línea]. Available: http://www.guiadigital.gob.cl/guiaweb_old/guia/archivos/Capitulo_III.pdf. [Último acceso: 2016 11 29].
- [80] I. J. G. B. James Rumbaugh, El Lenguaje Unificado de Modelado. Manual de Referencia, Madrid: Grafilles,S.L, 2000.
- [81] I. O. f. S. (ISO), 01 09 2013. [En línea]. Available: http://homes.ieu.edu.tr/userhat/courses/SE344_Fall_2015-2016/ISO_IEC_IEEE_29119_Part3.pdf. [Último acceso: 29 11 2016].
- [82] E. Serna y A. Serna, «Una Evaluación de las herramientas libres para pruebas de software,» *Redalyc*, nº 37, pp. 44-61, 2012.
- [83] S. M. Fariña, 15 06 2011. [En línea]. Available: <http://upcommons.upc.edu/bitstream/handle/2099.1/12515/73921.pdf>. [Último acceso: 12 04 2016].
- [84] A. d. F. Jiménez, 17 07 2015. [En línea]. Available: http://oa.upm.es/38561/1/TFG_Adri%C3%A1n_de_Frutos_Jim%C3%A9nez.pdf. [Último acceso: 04 12 2016].
- [85] H. A. P. Gélvez, 2010. [En línea]. Available: http://oa.upm.es/4019/1/HUGO_ALEXER_PARADA_GELVEZ.pdf. [Último acceso: 29 10 2016].
- [86] E. G. d. C. Glez, 14 01 2014. [En línea]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/26821/6/evagonzalezTFM0114memoria.pdf>. [Último acceso: 14 12 2016].
- [87] E. Y. Cedeño, 2015. [En línea]. Available: http://www.dit.upm.es/~posgrado/doc/TFM/TFMs2014-2015/TFM_Ericka_Yanez_Cedeno_2015.pdf. [Último acceso: 25 05 2017].
- [88] N. Yuquilema, 2016. [En línea]. Available: <http://dspace.unach.edu.ec/bitstream/51000/1501/1/UNACH-EC-ISC-2016-0014.pdf>. [Último acceso: 25 05 2017].
- [89] OWASP, 2013. [En línea]. Available: https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf. [Último acceso: 25 05 2017].
- [90] J. C. G. Ortiz, 16 12 2015. [En línea]. Available: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/10944/1/Gonz%C3%A1lez%20Ortiz,%20Juan%20Carlos.pdf>. [Último acceso: 16 12 2016].
- [91] C. M. B. A. S. R. y. V. S. Francisco Javier Díaz, 2007. [En línea]. Available: http://www.linti.unlp.edu.ar/uploads/docs/evaluacion_de_herramientas_open_source_para_pruebas_de_software.pdf. [Último acceso: 16 12 2016].

- [92] C. D. J. C. VELÁSQUEZ, 04 06 2009. [En línea]. Available: http://www.bdigital.unal.edu.co/930/1/8357252_2009.pdf. [Último acceso: 16 12 2016].
- [93] I. Institute of Electrical and Electronics Engineers, 16 09 1998. [En línea]. Available: <http://sci-hub.bz/10.1109/IEEESTD.1998.88820>. [Último acceso: 29 11 2016].
- [94] S. a. S. E.-S. testing, 01 09 2013. [En línea]. Available: http://homes.ieu.edu.tr/userhat/courses/SE344_Fall_2015-2016/ISO_IEC_IEEE_29119_Part3.pdf. [Último acceso: 29 11 2016].
- [95] I. S. T. Q. B. (ISTQB), 2016. [En línea]. Available: <http://www.istqb.org/downloads/syllabi.html>. [Último acceso: 30 11 2017].
- [96] J. J. G. Rodríguez, Noviembre 2011. [En línea]. Available: http://fondosdigitales.us.es/media/thesis/1724/O_Tesis-PROV18_.pdf. [Último acceso: 23 10 2016].
- [97] P. P. Dasiewicz. [En línea]. Available: <http://sci-hub.bz/10.1109/CCECE.2002.1013026>. [Último acceso: 29 10 2016].
- [98] C. M. Overstreet. [En línea]. Available: <http://sci-hub.bz/10.1109/WSC.2002.1172942>. [Último acceso: 29 10 2016].
- [99] M. E. M. M. y. A. R. Javier Gutiérrez, 2006. [En línea]. Available: <http://users.dsic.upv.es/workshops/dsdm06/files/dsdm06-07-Gutierrez.pdf>. [Último acceso: 29 10 2016].
- [100] J. M. S. Peño, 06 2015. [En línea]. Available: http://oa.upm.es/40012/1/PFC_JOSE_MANUEL_SANCHEZ_PENO_3.pdf. [Último acceso: 18 11 2016].
- [101] V. R. O. Jordán Enriquez Odalys, «redalyc.org,» 2006. [En línea]. Available: <http://www.redalyc.org/pdf/3604/360433560007.pdf>. [Último acceso: 29 10 2016].
- [102] D. G. M. D. Dalila Jústiz, «scielo.sld.cu,» 11 04 2013. [En línea]. Available: <http://scielo.sld.cu/pdf/rii/v35n2/rii03214.pdf>. [Último acceso: 29 10 2016].
- [103] F. J. P. J. M. M. Martha Lucia Rojas, «Facultad de Ingeniería,» 10 05 2015. [En línea]. Available: <http://revistas.uptc.edu.co/revistas/index.php/ingenieria/article/view/3551/3175>. [Último acceso: 25 10 2016].

11. Anexos

Anexo 1: Propuesta de la realización del Trabajo de Titulación en la UTI



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES


Ingeniería en Sistemas

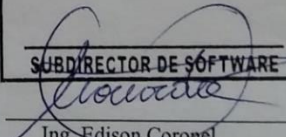
PROPUESTA


En reunión realizada el día Miércoles 27 de Abril del 2016 conjuntamente con el Director de tesis Ing. Hernán Torres, Director de la UTI Ing. Milton Labanda y Subdirector Ing. Edison Coronel y en presencia de la Tesista Lady Iñiguez, se revisó los objetivos y el alcance del proyecto denominado **“Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.”** el mismo que fue acogido y de mucho interés para las autoridades de la UTI; enfatizando que la mejora, se plantea para el área de Aseguramiento de la Calidad del Proceso y del Producto (PPQA) en el departamento de software, luego de esta presentación los Directivos de la UTI estuvieron de acuerdo y se comprometieron a brindar todo el apoyo para el desarrollo del presente proyecto.

Quienes a continuación firmamos estamos de acuerdo con el desarrollo del proyecto y nos comprometemos a brindar el apoyo necesario para su continuación.


Ing. Milton Labanda
Director de Telecomunicaciones e Información UNL



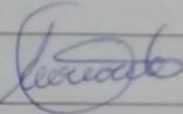

Ing. Edison Coronel
Subdirector de Desarrollo de Software


Ing. Hernán Torres
Director de Trabajo de Titulación


Lady Iñiguez
Tesista



Anexo 2: Entrevistas

ENTREVISTA # 1	
ENTREVISTADO: Ing. Edison Coronel CARGO: Subdirector de Desarrollo de Software REALIZADA POR: Lady del Cisne Ñiguez Carchi	Firma:  FECHA: 20/05/2016
<p>1. ¿Cuántas personas trabajan actualmente en el departamento de desarrollo de software de la UTI? 8 personas, actualmente la Lic. Mabel Rodríguez es responsable de la Seguridad de la Información, el Ing. José Martínez es responsable de Base de Datos, Ing. Verónica Gonzales, Ing. Jorge Malla, Ing. Vladimir Romero, Ing. Mario León, Ing. Lissette López tienen la responsabilidad de Analistas y desarrolladores.</p> <p>2. ¿En el desarrollo de un proyecto software elaboran una planificación y que toman en cuenta para la planificación? ¿Cuáles son las fases para la planificación? Exactamente todo proyecto que se realiza en la Universidad Nacional de Loja tiene que estar planificado en base al Plan Operativo Anula (POA), para que sean aprobados y para que luego tengan su financiamiento correspondiente, siendo evaluado semestralmente. Las fases para planificar son: Determina primeramente un Jefe de Proyecto el cual verificara o está a cargo de todas las fases del mismo. Definir la magnitud del proyecto para poder realizar la inversión que conlleva desarrollar el mismo un claro ejemplo es el SIAA (Sistema Informático Académico Administrativo) el cual aún no está aprobado por falta de presupuesto. El mismo que es la columna vertebral de todos los sistemas de la Universidad, es el que va a reemplazar al SGA actual. Tomando en cuenta es se debe planificar o adecuarse a las fases del ciclo de vida de un proyecto en general. Especificación de requerimientos en el cual se involucra al cliente determinando con los mismos que es lo que se quiere, revisando la parte legal para poder implementar la solución. Análisis en esta fase el Jefe de Proyecto realiza un análisis basándose en la estructura medular del mismo, situando toda esa información en una herramientas colaborativas como es el caso de Gitlab para poder realizar la maquetación del proyecto conjuntamente con los integrantes encargados del proyecto. Diseño en esta fase se definir cómo va ser estandarizado y cada uno de los técnico tiene que basarse en los estándares que se quede de acuerdo como uso de herramientas, colores es decir todo lo que inmiscuye prototipo de pantallas. Desarrollo se realiza la programación con las herramientas e información básica prevista en la etapa de Diseño. Capacitación del usuario final por parte de los miembros del GDS que estuvieron inmiscuidos en el proyecto. Puesta en producción es la parte donde se realiza la verificación del software, realizando las pruebas pertinentes, haciendo uso de los equipos que actualmente cuanta la UTI.</p>	

3. ¿Se involucra a los clientes en el proceso de desarrollo de un producto software?

Si se hacen un trabajo conjunto con el encargado o cliente que solicita el proyecto un claro ejemplo es el proyecto Distributivo Docente ahí se trabajó con Coordinación de Docencia determinando que es lo que se quiere se revisa la parte legal y se implementa la solución.

**4. ¿Dispone de manuales de políticas y normas de la calidad del producto software?
¿Qué políticas de calidad maneja?**

En la actualidad no existe ningún tipo de políticas o procedimientos de la calidad lo único que existe en la UTI es un normativo de políticas pero no abarca todos los campos de la UTI, está incompleto y para la parte de la calidad de software no existe nada.

5. ¿Los procesos realizados por el equipo de desarrollo software son verificados por usted? ¿Cómo es el proceso para el desarrollo de un proyecto software paso a paso?

Si son revisados paso a paso haciendo uso de herramientas colaborativas, en caso que algún miembro del GDS realice un cambio lo sube para que sea aprobado, teniendo una cuenta principal administrativa que está bajo la Subdirección, una vez que esté aprobado se hace una petición formal para el paso de producción una vez que ya este sin fallos, habiendo realizado las pruebas respectivas y a partir de ahí ya se clona a los respectivos servidores de producción.

6. ¿Existen personas encargadas de llevar las actividades de aseguramiento de la calidad de software? ¿Quiénes?

Por el momento no lo que se hace ahora es entre nosotros mismo verificarnos si un técnico realizo un sistema pues se lo pasa a otro técnico que no ha estado involucrado en el proyecto para que verifique que cumple con lo seguridades básicas.

7. ¿Qué herramientas conoce para verificar el aseguramiento de la calidad?

Conozco herramientas para llevar el control y diseño de las fases de un proyecto o para la usabilidad y accesibilidad pero realmente para aseguramiento de la calidad no.

8. ¿Considera que el departamento de desarrollo de software requiere personal dedicado al aseguramiento de la calidad?

Pues sí, pero lamentablemente el presupuesto es el limitante, inclusive para el pedido de personal en este año se sugirió personal para el aseguramiento de la calidad pero como el presupuesto de la Universidad fue recortado pues no se pudo tener ese personal.

ENTREVISTA # 2

ENTREVISTADO: Ing. Vladimir Romero
CARGO: Analista/Desarrollador
REALIZADO POR: Lady del Cisne Iñiguez Carchi

Firma:

FECHA: 24/05/2016

1. ¿Cuál es el rol que desempeña dentro del GDS?

Desarrollador y Analista.

2. ¿Cuáles son sus responsabilidades?

Como Desarrollador, ejecutar las funcionalidades del módulo del cual estoy a cargo el cual es Manejo de Bodegas todo lo que tiene que ver con bienes, ingresos, egresos, subida de matrices al Ministerio de Finanzas, todo ese tipo de cosas el mismo que debe luego integrarse con el SIAA.

Como Analista, levantar los procesos, modelar los diagramas de clase, realizar los modelos conceptuales y documentando.

3. ¿Con que metodología se trabaja?

Actualmente no se tiene una metodología base para cada fase del ciclo de vida de un proyecto pero tratamos de acoplarnos a la metodología de trabajo SCRUM.

4. ¿Qué herramientas utiliza en el entorno de desarrollo?

El lenguaje en el que se desarrolla es Python, como Framework Django, Angular y como Base de datos PostgreSQL.

5. ¿En el desarrollo de un proyecto software elaboran una planificación? ¿Cuáles son las fases para la planificación?

Previamente se realiza un análisis de factibilidad, puesto que por lo general hay departamentos que solicitan varios sistemas entonces lo que se procede hacer es trabajar con el sistema de mayor prioridad todo esto quedando de acuerdo con el cliente, luego de esto se realiza el análisis con los requisitos obtenidos previamente para poder obtener la debida aprobación, seguido del diseño y al desarrollo en sí.

6. ¿Cómo se realizan las pruebas del producto software?

En este caso se aplica lo que es pruebas cruzadas pues si se desarrolla una funcionalidad X se le atribuye la misma a otro desarrollador para que realice las pruebas. También se tiene las pruebas de aceptación en este caso ya es con el usuario final.

7. ¿Cómo se realiza el reporte y seguimiento de errores?

Básicamente no existe reporte alguno puesto que las pruebas se las hacen estando el desarrollador presente manejando una matriz especificada la funcionalidad que se ha cumplido o no como también la prioridad.

8. ¿Qué herramientas conoce para verificar el aseguramiento de la calidad?

Ninguna

ENTREVISTA # 3

ENTREVISTADO: Ing. Jorge Malla

CARGO: Analista/Desarrollador

REALIZADO POR: Lady del Cisne Iñiguez Carchi

Firma:

FECHA: 24/05/2016

1. ¿Cuál es el rol que desempeña dentro del GDS?

Analista y Desarrollador.

2. ¿Cuáles son sus responsabilidades?

Como Desarrollador, estoy a cargo del EVA básicamente estoy haciendo un script en el cual se pueda migrar la información de los docente, estudiantes y los planes de estudio del SGA al EVA con la finalidad de que un estudiante se enrole en el EVA y pueda ver sus cursos y las unidades que recibe. También estoy a cargo de un Sistema de Planificación y Desarrollo siendo un módulo del SIAA

Como Analista, Analizar procesos, los nuevos framework todo lo que tiene que ver con nuevas tecnologías para poder aplicar al SIAA. Entregando la documentación semanal de los avances que hemos tenido.

3. ¿Con que metodología se trabaja?

Pues en realidad no se sigue una metodología en concreto pero estamos utilizando ciertas fases de SCRUM y XP.

4. ¿Qué herramientas utiliza en el entorno de desarrollo?

GLPI es una herramienta que nos permite distribuir actividades a cada uno, también se trabaja con ODOO se sube las actividades de los sistemas que se está desarrollando ayudando mucho para generar la documentación.

5. ¿En el desarrollo de un proyecto software elaboran una planificación? ¿Cuáles son las fases para la planificación?

Se elabora una planificación estimando el tiempo en base a nuestra experiencia, basándose en el documento Especificación de requisitos IEEE 830 para el levantamiento de requerimientos, así como con el modelo 4+1 en el cual tenemos lo que es casos de uso, modelos conceptuales, diagrama de clases y diagramas de actividades.

6. ¿Cómo se realizan las pruebas del producto software?

Tenemos 3 ambientes un ambiente de desarrollo local en nuestras computadoras, ambiente de capacitación al usuario el cual hace las respectivas pruebas determinado si está de acuerdo o no con lo desarrollado y por último el ambiente de producción donde el tester que es un integrante del GDS realiza las pruebas y determina que el software cumple con lo establecido.

7. ¿Cómo se realiza el reporte y seguimiento de errores?

En este momento aún estoy en la etapa de desarrollo, al único que se le está realizando pruebas es al EVA donde por medio de correo electrónico o GLPI me hacen llegar las diferentes dificultades por ejemplo un Administrador no tiene todos los permisos o cosas así, los cuales ya fueron resueltos.

8. ¿Qué herramientas conoce para verificar el aseguramiento de la calidad?

Ninguna.

ENTREVISTA # 4

ENTREVISTADO: Ing. Verónica González
CARGO: Analista/Desarrollador
REALIZADO POR: Lady del Cisne Iñiguez Carchi

Firma:

FECHA: 24/05/2016

1. ¿Cuál es el rol que desempeña dentro del GDS

Analista de Sistemas

2. ¿Cuáles son sus responsabilidades?

Analista de Sistemas es la nominación que tengo dentro de la Universidad pero las funciones que se cumplen como tal son Análisis y Desarrolladora de software.

Como Analista, realizo el soporte a usuarios en los sistemas SGA Estudiantes, SGA Docentes, SGA Administrativos, también se tiene otro tipo de soportes como manejar o gestionar los requerimientos dentro de la mesa de ayuda de la Universidad Nacional de Loja que esta soportada por el sistema Gestión de Incidentes (GLPI) y también brindo soporte a la plataforma de la página web de la Universidad nacional de Loja.

Como desarrolladora, pues ya es en si el desarrollo de la aplicación o realizar los cambios de mejora que se tenga que hacer a los diversos sistemas, ahora estoy encargada de la integración del Sistema de Talento Humano interno de la Universidad con el Sistema de Talento Humano del Ministerio de Relaciones Laborales.

Y aparte de estas responsabilidades también cumplo la función de ser como encargada o tutora de algunos tesista para el tema de Mejoramiento y Diseño de Procesos, que tiene que incluirlo dentro de sus temas de tesis. También tengo la responsabilidad de delegar las funciones de los estudiantes que hacen prácticas preprofesionales de la Carrera de Sistemas.

3. ¿Con que metodología se trabaja?

SCRUM, no está formalizada pero esa es la metodología que se utiliza.

4. ¿Qué herramientas utiliza en el entorno de desarrollo?

Python es el lenguaje en que se desarrolla toda aplicación el mismo que se puede combinar con varias framework o algunas herramientas adicionales como Angular, Django y Bootstrap los que nos permite tener una mejora en la interfaz.

5. ¿En el desarrollo de un proyecto software elaboran una planificación? ¿Cuáles son las fases para la planificación?

Realmente en sí es el análisis preliminar que se hace, reunirse en grupo y hacer el diseño del mapa conceptual, al utilizar Python nuestra base es el modelo conceptual que es como el modelo en sí de cómo debería interactuar la aplicación, se discute las posibles soluciones que podemos dar. Luego se define en que lenguaje se va diseñar o como se va diseñar y poder cumplir con todos los requerimientos porque en realidad el usuario hace un pedido muy general, más bien luego de definir todo nos reunimos con el usuario y les hacemos conocer que opciones va tener el sistemas.

6. ¿Cómo se realizan las pruebas del producto software?

Las pruebas se realizan pidiéndonos apoyo mutuo por ejemplo yo no me puedo testear lo que yo mismo hago entonces le pido a algún compañero que me ayude testeando.

7. ¿Cómo se realiza el reporte y seguimiento de errores?

Primero que nada no hay un procedimiento formalizado de lo que es el desarrollo, se propuso un diagrama de proceso de cómo debería funcionar pero eso no está formalizado y no está implementado, entonces realmente estamos trabajando a lo que mejor se puede con lo que creemos que es mejor, lo mejor sería tener una herramienta que permita hacer seguimiento pero por ahora se realiza un reporte oral estando yo presente y mi compañero me dice ese momento en que está fallando el software.

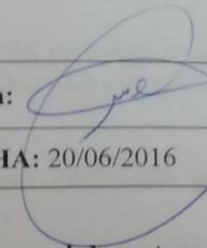
8. ¿Qué herramientas conoce para verificar el aseguramiento de la calidad?

Más que una herramienta lo que conozco es la metodología de hacer pruebas de caja negra, pruebas de caja blanca, pruebas unitarias, pruebas de ciclos, stress, carga, pero herramienta en sí no.

ENTREVISTA # 5

ENTREVISTADO: Ing. Milton Labanda
CARGO: Director de Telecomunicaciones e Información
REALIZADO POR: Lady del Cisne Iñiguez Carchi

Firma:



FECHA: 20/06/2016

1. Cree usted que se debería implementar mejora de procesos en el departamento de software?

Si porque nos hace falta mecanismos o procedimientos para ordenar y organizar el trabajo que se hace, un trabajo de valor y de mucha trascendencia para la Universidad en especial en estos últimos meses pues si se necesita procesos de este tipo que aseguren la calidad.

2. Que beneficios tendría el departamento al implementar mejora de procesos software.

Pienso que apostaría en cuanto a poder medir y controlar la calidad del trabajo que se está haciendo.

3. En el departamento de software tiene gente especializada para cada etapa de un proyecto software.

Realmente por la demanda de trabajo, demanda de operaciones, trabajo de soporte, por la intensidad de los procesos no; si se tiene gente especializada con los perfiles pero realmente no se puede implementar los procesos en sí, prácticamente todos hacen de todo bajo la demanda de la elevada carga de soporte que hay que hacer a la mayoría de sistemas.

4. ¿Considera que el departamento de desarrollo de software requiere personal dedicado al aseguramiento de la calidad?

Claro que si requiere, es muy justificado principalmente por el hecho de que necesitamos medir el producto que se está desarrollando y la calidad del mismo.

5. ¿Dispone de manuales de políticas y normas de la calidad del producto software?

Existen normas aprobadas por la anterior administración de la unidad, donde escasamente se toca el tema de calidad, pero nosotros actualmente estamos actualizando esas políticas y creando nuevas pues aún estamos en fase de revisión. Dentro de las cuales se tiene ciertos lineamientos generales pero la visión que tenemos actualmente es que a todas esas políticas ir las aprobando como primera instancia y hacerlas como una base después las iremos detallando a medida que avanza el tiempo.

6. ¿Los procesos realizados por el equipo de desarrollo de software son verificados por el Subdirector de Desarrollo de Software?

Teóricamente si, en general el hace la revisión cualquier notificación se la realizan a él y en ocasiones a mí, pero como él es el especialista y como subdirector pues tiene el perfil más óptimo para hacer las revisiones.

7. La UTI tiene algún convenio con empresas o instituciones nacionales para desarrollar productos software.

No, únicamente interno a la Universidad si tenemos esa proyección pero lógicamente cuando sentemos una parte de procesos y herramientas si se quiere base para que funcione aquí en la Universidad pues ahí sí podría producir hacia afuera.

8. la UTI tiene algún convenio con empresas o instituciones nacionales para desarrollar productos software.

No, lo único que existe son convenios para pasantías no más para desarrollar software de igual manera se tiene la perspectiva de poder cambiar productos o desarrollo con Universidades o Instituciones de afuera.

Anexo 3: Políticas de la UTI



UNIVERSIDAD NACIONAL DE LOJA

ACREDITADA-Resolución N°003-CONEA-2010-111-DC

020131375

Of. Nro. -SG-UNL
Loja, 10 de Mayo de 2013

Trámite Nro. 191965

Señores:

VICERRECTOR.
DIRECTORES DE LAS AREAS Y DE LA MED.
DIRECTOR DE TELECOMUNICACIONES E INFORMACION.
DIRECTOR FINANCIERO.
DIRECTOR ADMINISTRATIVO.
PROCURADOR.
AUDITORA.
Ciudad.

De mi consideración:

Para su conocimiento y fines consiguientes, adjunto un ejemplar de la Resolución Nro. 019/2013-R-UNL, de 9 de Mayo de 2013, emitida por el Dr. Gustavo Enrique Villacís Rivas, Rector de la Institución, que determina sobre "LAS POLITICAS DE TELECOMUNICACIONES, DESARROLLO DE SOFTWARE, REDES" DE LA UNIVERSIDAD NACIONAL DE LOJA.

Me permito solicitar a Usiedes, se sirvan difundir la presente resolución a las secciones o departamentos correspondientes, y se publique en la página web de la Universidad Nacional de Loja.

Con los sentimientos de mi especial consideración.

Muy atentamente,
EN LOS TESOROS DE LA SABIDURIA
ESTA LA GLORIFICACION DE LA VIDA

Dr. Ernesto Roldán Jara,
SECRETARIO GENERAL



c.c. Directora de Bienestar Estudiantil,
Secretaría General,
Archivo.

ERJ/nlmdeA.

Ing. Milton Labandey



sistema de registro de solicitudes de los usuarios de la institución, con el fin de medir el trabajo realizado y la eficiencia del servicio brindado.

2. POLÍTICAS ESPECÍFICAS A LAS SECCIONES DE LA UTI.

POLITICAS DE LA SECCIÓN DE DESARROLLO DE SOFTWARE

2.1. Políticas de proyectos.

Los proyectos que la Universidad Nacional de Loja aborde, y que se relacionen con la tecnología de información, deberán estar contemplados en el Plan de Desarrollo de la UTI.

El Plan de Desarrollo de la UTI de la Universidad Nacional de Loja, y por lo tanto el plan de desarrollo de sistemas incluido en él, deberán estar autorizados por las máximas autoridades de la Universidad o por la estructura administrativa que sea establecida para tales efectos.

El Plan de Desarrollo deberá ser revisado con una periodicidad mínima de una semana en cuanto a cumplimiento y orientación.

El Plan de Desarrollo de la UTI deberá contemplar, al menos, los siguientes puntos:

- Plan de Sistemas
- Plan de Hardware
- Plan de Software
- Plan de Telecomunicaciones

En los contratos de desarrollo de sistemas, que se convengan con empresas consultoras externas, se incluirá una garantía de cumplimiento de lo convenido contractualmente, por un período de a lo menos un (1) año a contar de su implantación.

2.2 Políticas de metodologías de desarrollo.

El desarrollo de sistemas se llevará a cabo mediante el uso de técnicas estructuradas de análisis y diseño de sistemas.



2.5. Políticas de explotación de sistemas.

La explotación de los sistemas de información residentes en los computadores de la UTI, será de exclusiva responsabilidad de los administradores de los mismos.

La UTI será el organismo encargado de proporcionar o coordinar la capacitación de los usuarios para la incorporación, operación y uso de sistemas de información.

Los funcionarios que tengan equipos, serán responsables de la conservación y buen uso de dicho equipamiento.

2.6. Políticas de desarrollo de sistemas y aplicaciones en computadores.

Las funciones, procesos, sistemas y aplicaciones que se realicen con los productos de computación, deberán constituirse en parte integral de los sistemas de la Universidad Nacional de Loja, como complemento a los sistemas y/o funciones de ésta y no transformarse en sustitutos de ellos.

La protección, custodia y respaldo de los datos pertenecientes a los sistemas y aplicaciones basados en computadores y que no se tengan en el servidor de la red, serán de responsabilidad exclusiva de sus administradores o usuarios.

La UTI será el organismo encargado de proporcionar apoyo y asesoría técnica a los usuarios para la incorporación y uso de productos estándares de computación que requieran.

3. POLÍTICAS, NORMAS Y PROCEDIMIENTOS PARA LA ELABORACIÓN DE SISTEMAS.

Objetivo.

Establecer las políticas, normas y procedimientos que permitan homogeneizar el desarrollo de sistemas de información dentro de la Universidad Nacional de Loja.



Introducción.

El software es el objeto de análisis y estudio de una de las ramas más jóvenes de ingeniería. De hecho, podemos decir que la concepción misma del software, es decir, su naturaleza de construcción y desarrollo es lo que lo hace particularmente distinto. Entre las características fundamentales del software podemos mencionar que su naturaleza es abstracta debido a que su herramienta principal es el manejo de la información. Además el software se desarrolla, no se fabrica como cualquier producto en un sentido clásico; se crea mediante la transformación del poder intelectual y cerebral de los especialistas en el conocimiento.

Consecuentemente, siendo la información uno de los recursos más valiosos de cualquier institución u organización, es de gran importancia contar con sistemas que permitan su uso eficiente, en armonía con todas las áreas. Para garantizar que esto ocurra dentro de la Universidad Nacional de Loja es indispensable contar con políticas, normas y procedimientos que permitan homogeneizar la elaboración de sistemas de información.

3. 1. Políticas para la elaboración de sistemas.

La política de calidad se define como las directrices u objetivos generales que tiene la Universidad Nacional de Loja concernientes a la calidad, las cuales son emitidas por la Unidad de Telecomunicaciones e Información.

Objetivo.

Establecer un marco de referencia general para garantizar que los sistemas de información que apoyan las tareas sustantivas de una institución o empresa, sean concebidos y desarrollados de una manera tal, que permitan su articulación para una adecuada interacción entre las áreas.

Políticas



UNIVERSIDAD NACIONAL DE LOJA

ACREDITADA-Resolución N°003-CONEA-2010-111-DC

- Toda elaboración de sistemas deberá estar orientada a satisfacer las necesidades de manejo de información para las funciones sustantivas de la Universidad Nacional de Loja; es importante concebir el diseño de dichos sistemas de manera que permitan su integración y consolidación en una base de datos.
- Toda elaboración de sistemas, tanto interna como externa, debe cumplir con las normas establecida por la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, El cumplimiento de las normas es un requisito indispensable para considerar un sistema apto para su liberación definitiva.
- Toda elaboración de sistemas, tanto interna como externa de carácter institucional, deberá estar avalada por un informe técnico de la unidad de Telecomunicaciones e Información, este debe normar el uso y aprovechamiento de los recursos informáticos de acuerdo al reglamento interno de la Institución.
- La elaboración de sistemas institucionales debe apegarse a los estándares en cuanto al uso de software. Cuando esto no sea posible, el área usuaria deberá solicitar un dictamen técnico a la dirección de informática de la institución o empresa, justificando plenamente el uso de las herramientas propuestas para el desarrollo.
- Todos los sistemas y sus componentes desarrollados por el personal de la institución son propiedad de la Institución, por lo que la Universidad tendrá los derechos de autor para la utilización de dichos desarrollos en las diferentes áreas que así lo requieran.
- Durante el análisis, desarrollo e implantación de cualquier sistema, el área solicitante deberá participar con la Unidad de Telecomunicaciones e Información.
- Es responsabilidad de la Unidad de Telecomunicaciones e Información, el proporcionar la capacitación y asistencia técnica al personal operativo sobre el correcto uso del Sistema.
- La Unidad de Telecomunicaciones e Información establecerá de manera formal su política de calidad en cuanto a las normas y procedimientos por utilizar, con objeto de que funcione eficazmente el sistema informático de la Institución.
- En la elaboración y diseño de sistemas informáticos internos la Unidad de Telecomunicaciones e Información será la encargada



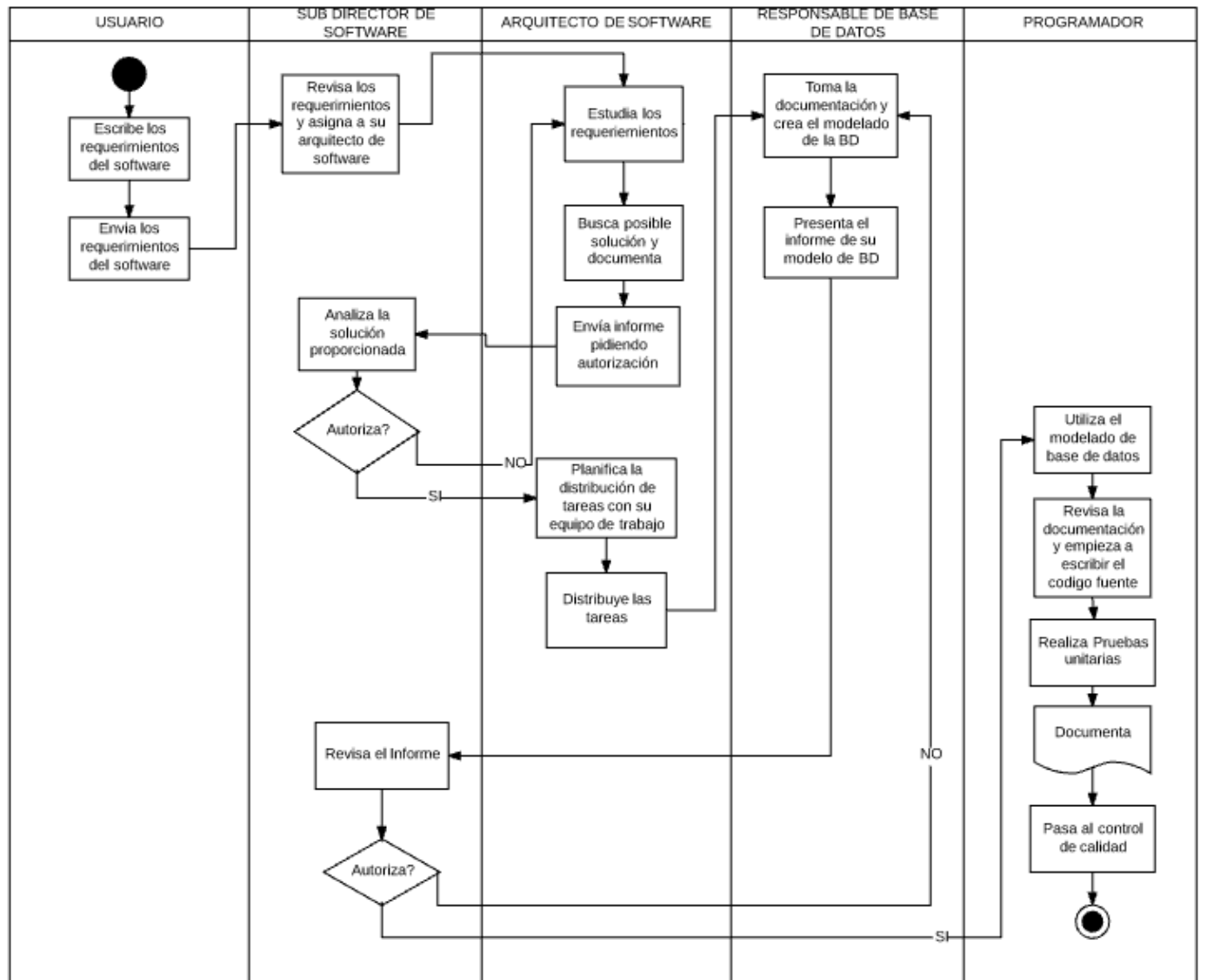
3.5. Normas para la programación y documentación de sistemas.

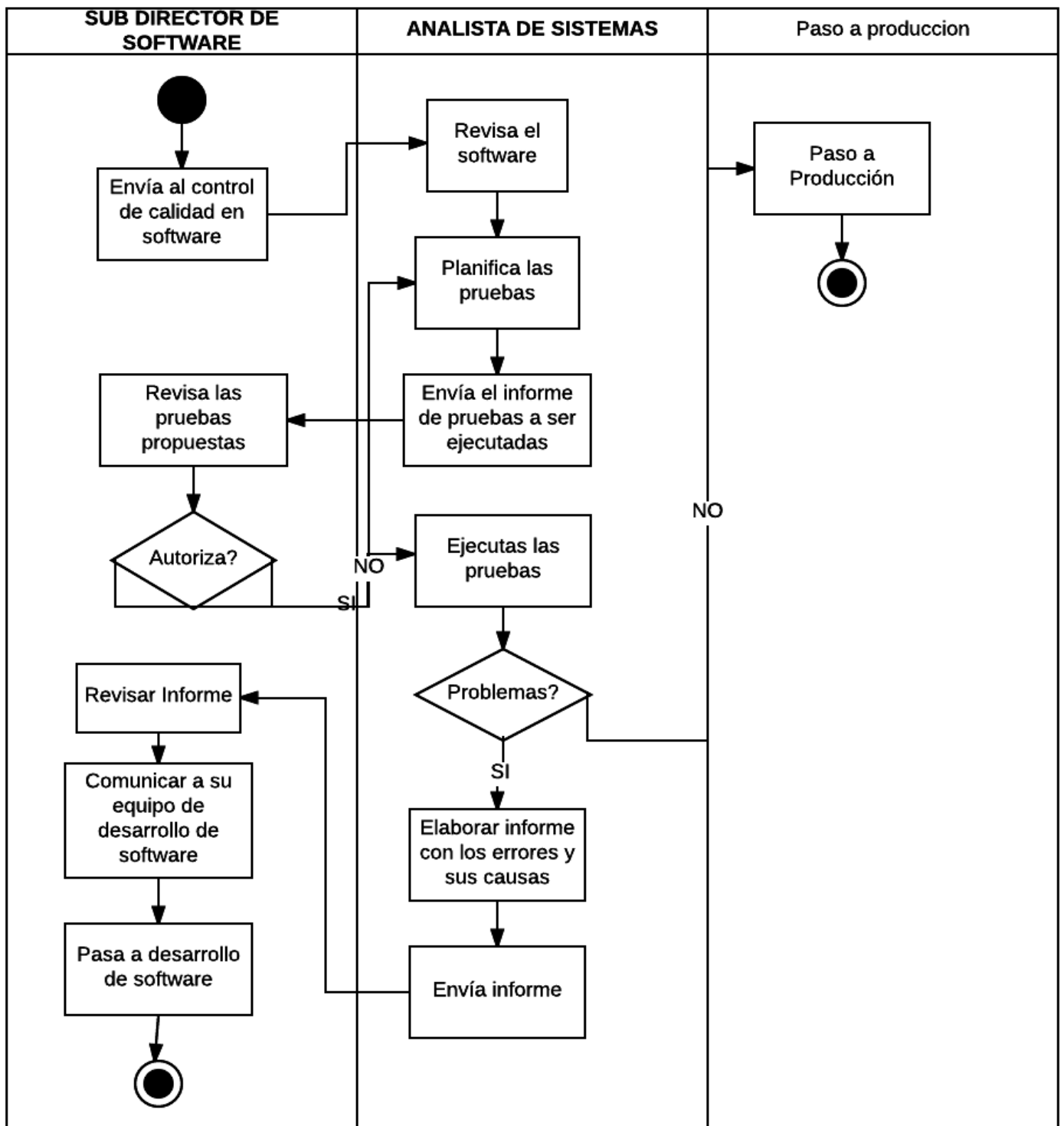
1. Todos los programas que integren cualquier sistema deberán estar documentados conforme al Manual de Procedimientos para el Desarrollo de Sistemas.
2. El área usuaria deberá aprobar el manual del usuario previo a la liberación de un sistema. La Unidad de Telecomunicaciones e Información deberá revisar que el manual técnico se apegue a las especificaciones.
3. La Unidad de Telecomunicaciones mantendrá un control de la documentación de los sistemas.
4. La Unidad de Telecomunicaciones tendrá un estricto control de documentación, actualizando los contenidos según el mejoramiento continuo de los sistemas.
5. Todos aquellos códigos que sean objeto de programación, ya sean módulos, programas, pantallas, etc., deberán contener información de quién efectuó la programación y en qué fecha; de ser posible en el mismo software, mediante comentarios y adicionalmente en la documentación por escrito.
6. Después de concluida la programación de una parte del sistema, se deberá registrar en un documento que dicha parte del sistema ha sido concluida, especificar el o los nombre(s) del o los programador(es), así como el tiempo de programación en horas; esto con el fin de establecer un control de calidad del trabajo de los programadores.

3.6. Normas para la implantación de sistemas y capacitación.

1. Antes de liberar un nuevo sistema, éste deberá ser sometido a pruebas de aceptación definidas por el área solicitante, utilizando para ello datos reales. En el caso de nuevas versiones, será necesario realizar corridas en paralelo para verificar su correcto funcionamiento con respecto a la versión anterior.
2. La capacitación al personal técnico-operativo formará parte fundamental de la liberación de un sistema. Dicha capacitación deberá cubrir todas las necesidades y requerimientos que el área usuaria especifique de común acuerdo con la Unidad de Telecomunicaciones e Información.

Anexo 4: Procesos actuales del GDS de la UTI





Anexo 5: Manual de Valoración de Puestos



UNIVERSIDAD NACIONAL DE LOJA

ACREDITADA-Resolución N°003-CONEA-2010-111-DC

Of. Nro. 0020142557 -SG-UNL
Loja, 1 de Agosto de 2014

Trámite Nro. 201668


Señores:
DIRECTOR FINANCIERO.
JEFA DE RECURSOS HUMANOS Y ESCALAFÓN.
Ciudad.

De mi consideración:

Para su conocimiento y fines consiguientes, adjunto un ejemplar de la Resolución Nro. 044-R-UNL-2014, de 31 de Julio de 2014, emitida por el Rectorado de la Institución, mediante la cual se aprueba el **MANUAL DE DESCRIPCIÓN, VALORACIÓN Y CLASIFICACIÓN DE PUESTOS DE LA UNIVERSIDAD NACIONAL DE LOJA.**

Con los sentimientos de mi especial consideración.

Muy atentamente,
EN LOS TESOROS DE LA SABIDURÍA
ESTA LA GLORIFICACIÓN DE LA VIDA


Dr. Ernesto Roldán Jara,
SECRETARIO GENERAL

c.c. Procuradora,
Auditora General,
Secretaría General,
Archivo.



ERJ/nlmdeA|

UNIVERSIDAD NACIONAL DE LOJA			
DESCRIPCIÓN Y PERFIL DE PUESTOS			
1. DATOS DE IDENTIFICACIÓN DEL PUESTO		4. RELACIONES INTERNAS Y EXTERNAS	5. INSTRUCCIÓN FORMAL REQUERIDA
Código:	03.01.02.19.3.08.02.10.0	INTERFAZ	
Denominación:	Subdirección de Desarrollo de Software	Coordinadora General Administrativa Financiera; Director de Telecomunicaciones e Información; autoridades académicas y administrativas; Dirección de Contratación Pública; Responsables de las Secciones de Gestión de Telecomunicaciones e Información; analistas y asistentes de la gestión de telecomunicaciones e información; usuarios internos; proveedores; personal de los organismos rectores de educación superior; Ministerio de Finanzas; SENPLADES y Ministerio de Relaciones Laborales.	Nivel de Instrucción:
Nivel:	Directivo		Tercer Nivel
Unidad o Proceso:	Gestión de Telecomunicaciones e Información		Título Requerido:
Rot:	Subdirección de unidad organizacional		SI
Grupo Ocupacional:	1-NJS		Área de Conocimiento:
Ámbito:	Local	Ingeniería en Sistemas Informáticos, y afines	
2. MISIÓN		6. EXPERIENCIA LABORAL REQUERIDA	
Planificar, coordinar y controlar los estudios de software necesarios para garantizar la eficacia, eficiencia y efectividad de los procesos académicos y administrativos de la Universidad Nacional de Loja.		Tiempo de Experiencia:	5 años
		Especificidad de la experiencia:	Desarrollo de Software, Sistemas Tecnológicos de Información y Telecomunicaciones; Sistemas Informáticos de Gestión Académica y Administrativa.
3. ACTIVIDADES ESENCIALES		7. CONOCIMIENTOS	
Planifica, coordina, controla y evalúa el diseño, implementación y mantenimiento de software requerido por las diferentes áreas académicas o unidades administrativas de la Universidad Nacional de Loja.		8. DESTREZAS / HABILIDADES	
Coordina, implementa y monitorea la optimización de los sistemas informáticos de gestión académica.		Diseño de Programas y Aplicaciones Informáticas; Sistemas Operativos; Normas de Control Interno; Sistemas Informáticos de Gestión Académica y Administrativa; Legislación Universitaria.	Diseño de tecnología
Coordina, implementa y monitorea la optimización de los sistemas informáticos de gestión administrativa y financiera.		Diseño de Programas y Aplicaciones Informáticas; Sistemas Operativos; Sistemas Informáticos de Gestión Académica; Administración de Bases de Datos.	Monitoreo y control
Coordina la preparación de las especificaciones técnicas de los pliegos para la adquisición o contratación de software especializado, en base a la planificación y proyectos aprobados por la Institución.		Diseño de Programas y Aplicaciones Informáticas; Sistemas Operativos; Legislación Universitaria; Sistema Nacional de Contratación Pública; Normas de Control Interno.	Diseño de tecnología
Establece y coordina la ejecución de las políticas de seguridad y respaldo de los sistemas informáticos de la institución.		Diseño de Programas y Aplicaciones Informáticas; Sistemas Operativos; Normas de Control Interno; Seguridad de Redes Informáticas y Software.	Pensamiento crítico
Coordina la capacitación a los usuarios internos de la operación del software adquirido o desarrollado por la Sección.		Sistemas Tecnológicos de Información y Telecomunicaciones; TICs; Sistemas Informáticos de Gestión Académica y Administrativa; Legislación Universitaria.	Instrucción

UNIVERSIDAD NACIONAL DE LOJA			
DESCRIPCIÓN Y PERFIL DE PUESTOS			
1. DATOS DE IDENTIFICACIÓN DEL PUESTO		4. RELACIONES INTERNAS Y EXTERNAS	5. INSTRUCCIÓN FORMAL REQUERIDA
Código:	03.01.02.19.3.08.01.02.1	INTERFAZ	
Denominación:	Analista de Sistemas Informáticos 1	Dirección de Telecomunicaciones e Información; Dirección de Contratación Pública; Responsables de las Secciones de Gestión de Telecomunicaciones e Información; analistas y asistentes de la gestión de telecomunicaciones e información; usuarios internos; proveedores; personal de los organismos rectores de educación superior; Ministerio de Finanzas; SENPLADES y Ministerio de Relaciones Laborales.	Nivel de Instrucción:
Nivel:	Profesional		Tercer Nivel
Unidad o Proceso:	Gestión de Telecomunicaciones e Información		Título Requerido:
Rot:	Ejecución de procesos		SI
Grupo Ocupacional:	Servidor Público 4		Área de Conocimiento:
Ámbito:	Local	Ingeniería en Sistemas Informáticos, y afines	
2. MISIÓN		6. EXPERIENCIA LABORAL REQUERIDA	
Ejecutar los procesos técnicos del sistema de telecomunicaciones e información de la Universidad Nacional de Loja, relacionados con desarrollo de software; diseño, implementación y mantenimiento de redes; y adquisición, renovación y mantenimiento de equipos informáticos.		Tiempo de Experiencia:	3-4 años
		Especificidad de la experiencia:	Sistemas Tecnológicos de Información y Telecomunicaciones; Sistemas Informáticos de Gestión Académica y Administrativa; Mantenimiento de Redes y Equipos Informáticos
3. ACTIVIDADES ESENCIALES		7. CONOCIMIENTOS	
Diseña, implementa y mantiene el software requerido por las diferentes áreas académicas o unidades administrativas de la Universidad Nacional de Loja.		8. DESTREZAS / HABILIDADES	
Ejecuta los procesos de optimización de los sistemas informáticos de gestión académica y administrativa.		Sistemas Informáticos de Gestión Académica y Administrativa; Administración de Bases de Datos.	Monitoreo y control
Realiza estudios, diseños, implementación y mantenimiento de redes informáticas, según los requerimientos académicos y administrativos institucionales.		Sistemas Tecnológicos de Información y Telecomunicaciones; Sistemas de Transmisión de Datos; Diseño, Instalación y Seguridad de Redes Informáticas; Normas de Control Interno.	Operación y control
Proporciona soporte técnico y mantenimiento preventivo y correctivo a las redes y equipos de los sistemas informáticos de la Universidad Nacional de Loja.		Conocimientos especializados de Hardware; Soporte Técnico y Mantenimiento de Equipos Informáticos; Electrónica.	Mantenimiento de equipos
Elabora las especificaciones técnicas de los pliegos para la adquisición o contratación de software especializado, equipos informáticos y materiales para su mantenimiento.		Sistemas Tecnológicos de Información y Telecomunicaciones; Sistema Nacional de Contratación Pública.	Pensamiento crítico
Capacita a los usuarios internos en la operación del software adquirido o desarrollado, equipos informáticos y TICs.		Sistemas Tecnológicos de Información y Telecomunicaciones; TICs; Normas de Control Interno; Sistemas Informáticos de Gestión Académica y Administrativa.	Instrucción



UNIVERSIDAD NACIONAL DE LOJA

DESCRIPCIÓN Y PERFIL DE PUESTOS

1. DATOS DE IDENTIFICACIÓN DEL PUESTO		4. RELACIONES INTERNAS Y EXTERNAS		5. INSTRUCCIÓN FORMAL REQUERIDA	
Código:	03.01.02.19.3.08.01.02.2	INTERFAZ		Nivel de Instrucción:	Tercer Nivel
Denominación:	Analista de Sistemas Informáticos 2	Directora de Telecomunicaciones e Información; Directora de Contratación Pública; Responsables de las Secciones de Gestión de Telecomunicaciones e Información; analistas y asistentes de la gestión de telecomunicaciones e información; usuarios internos; proveedores; personal de los organismos rectores de educación superior; Ministerio de Finanzas, SENPLADES y Ministerio de Relaciones Laborales.		Título Requerido:	SI
Nivel:	Profesional			Área de Conocimiento:	Ingeniería en Sistemas Informáticos; y afines.
Unidad o Proceso:	Gestión de Telecomunicaciones e Información.				
Rol:	Ejecución y supervisión de procesos				
Grupo Ocupacional:	Servidor Público 6				
Ámbito:	Local				
2. MISIÓN		6. EXPERIENCIA LABORAL REQUERIDA			
Ejecutar y supervisar los procesos técnicos del sistema de telecomunicaciones e información de la Universidad Nacional de Loja relacionados con desarrollo de software; diseño, implementación y mantenimiento de redes; y adquisición, renovación y mantenimiento de equipos informáticos.		Tiempo de Experiencia:	5-6 años		
		Especificidad de la experiencia:	Sistemas Tecnológicos de Información y Telecomunicaciones; Sistemas Informáticos de Gestión Académica y Administrativa; Desarrollo de Software; Hardware especializado; Mantenimiento de Redes y Equipos Informáticos.		
3. ACTIVIDADES ESENCIALES		7. CONOCIMIENTOS		8. DESTREZAS / HABILIDADES	
Ejecuta y supervisa el diseño, implementación y mantenimiento de software requerido por las diferentes áreas académicas o unidades administrativas de la Universidad Nacional de Loja.		Diseño de Programas y Aplicaciones Informáticas; Sistemas Operativos; Normas de Control interno; Legislación Universitaria.		Diseño de tecnología	
Ejecuta y supervisa los procesos de optimización de los sistemas informáticos de gestión académica y administrativa.		Sistemas Informáticos de Gestión Académica y Administrativa; Administración de Bases de Datos.		Monitoreo y control	
Ejecuta y supervisa los estudios, diseños, implementación y mantenimiento de redes informáticas, según los requerimientos académicos administrativos institucionales.		Sistemas Tecnológicos de Información y Telecomunicaciones; Sistemas de Transmisión de Datos; Diseño, Instalación y Seguridad de Redes Informáticas; Normas de Control Interno.		Operación y control	
Ejecuta y organiza el soporte técnico y el mantenimiento preventivo, correctivo a las redes y equipos de los sistemas informáticos de la Universidad Nacional de Loja.		Conocimientos especializados de Hardware; Soporte Técnico y Mantenimiento de Equipos Informáticos; Electrónica.		Mantenimiento de equipos	
Organiza y elabora las especificaciones técnicas de los pliegos para la adquisición o contratación de software especializado, equipos informáticos y materiales para su mantenimiento.		Sistemas Tecnológicos de Información y Telecomunicaciones; Sistema Nacional de Contratación Pública.		Pensamiento crítico	
Organiza y capacita a los usuarios internos en la operación del software adquirido o desarrollado, equipos informáticos y TIC's.		Sistemas Tecnológicos de Información y Telecomunicaciones; TIC's; Normas de Control interno; Sistemas Informáticos de Gestión Académica y Administrativa.		Instrucción	

Esc



Anexo 6: Aprobación de Procesos



UNIVERSIDAD NACIONAL DE LOJA

UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

SUBDIRECCIÓN DE DESARROLLO DE SOFTWARE

CERTIFICO:

Que la egresada Lady del Cisne Ñíguez Carchi ha realizado el **levantamiento de procesos de Aseguramiento de la Calidad del Software** como parte del proyecto de titulación *“Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.”* los mismos que han sido validados y aprobados por los Técnicos y Analistas que laboran en esta unidad y por mi persona, considerando que esta propuesta aportará a la mejora de los procesos de desarrollo de software y productos de software que se desarrollan en esta Unidad para el beneficio de la Universidad Nacional de Loja.



Ing. Edison Leonardo Coronel Romero

Subdirector de Desarrollo de Software
Unidad de Telecomunicaciones e Información

Anexo 7: Aprobación de Plantillas



UNIVERSIDAD NACIONAL DE LOJA

UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

UNIDAD DE DESARROLLO DE SOFTWARE

CERTIFICO:

Que la egresada Lady del Cisne Iñiguez Carchi ha realizado el **levantamiento de plantillas del aseguramiento de la Calidad del Software** como parte del proyecto de titulación *“Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.”* los mismos que han sido revisados y aprobados por mi persona, considerando que esta propuesta aportará a la mejora de productos software que se desarrollan en esta Unidad para el beneficio de la Universidad Nacional de Loja.

Ing. Lissette Geoconda López Faicán

Analista/Desarrollador

Anexo 8: Elección del estándar para las plantillas

Estándares para documentar un Plan de Pruebas

Para planificar o diseñar un plan de pruebas tenemos dos estándares que son los más utilizados y reconocidos IEEE 829-2008 y ISO/IEC/IEEE 29119-3:2013 e ISTQB. Se detallará las características ventajas y desventajas de cada uno para determinar cuál es el óptimo y tomar como base para el planteamiento del Plan de Pruebas en este trabajo en ejecución.

IEEE 829-2008

“El objetivo de esta norma es para describir un conjunto de documentos de pruebas de software básico. La definición del contenido de un documento de prueba estandarizado puede servir como una lista de comprobación de la integridad para el proceso de pruebas asociado. En muchas organizaciones, el uso de estos documentos aumenta significativamente la capacidad de administración de las pruebas. Este estándar especifica la forma y el contenido de cada uno de los documentos de prueba”. Sus ventajas y desventajas son: [93]

Ventajas

- Facilita la comunicación al proporcionar un marco común de referencia (por ejemplo, un cliente y un proveedor tienen la misma definición de un plan de prueba).
- Desglosa en 3 distintos tipos de documentos: Especificación de diseño de pruebas, Especificación de casos de prueba y Especificación de procedimientos de pruebas, haciendo su implantación ordenada y más entendible.
- Está permitido la libre elección por parte del Tester a la hora de seleccionar los documentos que se utilizará.
- Se puede optar por unir uno o varios documentos e incluir algún otro si fuere necesario.
- La norma IEE 829 ofrece ocho anexos que son de utilidad para realizar las pruebas, entre los cuales están: bibliografía, ejemplo de esquemas de niveles de integridad, tareas de prueba, ejemplos de documentación, etc.

Desventajas

- Es actualmente reemplazada por la norma ISO/IEC/IEEE 29119.

ISO/IEC/IEEE 29119-3:2013

“El objetivo de la norma ISO/IEC/IEEE 29119 es definir un conjunto de normas para las pruebas de software que puede ser usado por cualquier organización al realizar cualquier forma de testing de software, ISO/IEC/IEEE 29119-3:2013 incluye plantillas y ejemplos de documentos de prueba. La documentación de las plantillas definidas en la norma ISO/IEC/IEEE 29119-3:2013 se puede utilizar en conjunción con cualquier modelo de ciclo de vida de desarrollo de software”. Sus ventajas y desventajas son: [94]

Ventajas

- Especifica plantillas de documentación de prueba de software que puede ser usado por cualquier organización, proyecto o menor actividad de pruebas.
- Cada plantilla puede adaptarse a las necesidades únicas de cada organización que implementa el estándar, para apoyar la implementación del estándar dentro de cualquier modelo de ciclo de vida de desarrollo de software.
- El estándar de documentación de prueba IEEE 829, bien conocido y ampliamente utilizado, se utilizó como base para esta norma, y como tal, la norma ISO / IEC / IEEE 29119-3 reemplaza IEEE 829.
- No existe ningún requisito para los usuarios de este estándar para utilizar la misma estructura del documento o la misma nomenclatura (es decir, puede combinar o dividir documentos descritos en la norma).

Desventajas

- Consta de 3 tipos de documentación los mismos que se desglosan en 16 documentos en total, llegando a ser un trabajo que demanda de tiempo y recursos por parte de quien inicia la implantación de dicho estándar.

ISTQB

El Comité Internacional de Cualificación de Pruebas de Software (ISTQB) ha creado el esquema más exitoso del mundo para la certificación de los probadores de software, hasta junio de 2016, ISTQB ha administrado más de 650.000 exámenes y emitido más de 470.000 certificaciones en más de 110 países de todo el mundo.

La ISTQB pone a disposición de los probadores 3 tipos de documentos [95]:

- Foundation Level Syllabus: Sílabo de nivel básico que se constituye en la base para la calificación internacional de pruebas de software. Su objetivo es otorgar un entendimiento básico del proceso de pruebas de software.
- Advanced Level Syllabus: Este programa de estudio está destinado a personas que han alcanzado un estado avanzado en sus carreras en el ámbito de las pruebas de

software. El objetivo de este nivel es medir la coherencia entre la experiencia y ejecución de los proyectos de testing por parte del analista, este sílabo exige al menos 3 años de experiencia como tester.

- Expert Level Syllabus: Sílabo para probadores expertos que comprende una variedad de temas especializados, el cual exige para proceder a certificarse una experiencia práctica aprobada de al menos 5 años en el testing, y al menos 2 años de experiencia.

Dentro de Foundation Level Syllabus consta de un proceso básico de pruebas el que consta de las siguientes actividades: Planificación y control, análisis y diseño, implementación y ejecución, evaluación de los criterios de salida e informes y actividades de cierre de pruebas. Específicamente en la Planificación de pruebas da a conocer las actividades que debe incluir un Plan de pruebas.

Ventajas


- Las actividades especificadas en el plan de pruebas puede aplicarse a un sistema completo o parte de un sistema
- Las actividades definidas en el plan de pruebas se pueden acoplar a los distintos procesos de ciclo de vida del software

Desventajas

- Debe documentarse planes de prueba por separada para los distintos niveles de pruebas tales como pruebas de sistema y pruebas de aceptación
- El diseño del documento de plan de pruebas se basa en la norma IEEE 829-1998, el mismo que fue reemplazado por la ISO/IEC/IEEE 29119-3.

Luego del análisis realizado se decide trabajar con el estándar ISO / IEC / IEEE 29119-3:2013 el mismo que es uno de los estándares más usados a nivel mundial y más completos en cuanto a Testing, puesto que ISTQB se basa en los lineamientos de la IEEE 829 el mismo que fue reemplazando por el estándar ISO / IEC / IEEE 29119.

Anexo 9: Designación del Módulo de Planificación y Desarrollo Institucional para plan piloto

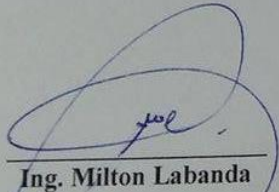



UNIVERSIDAD NACIONAL DE LOJA
UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

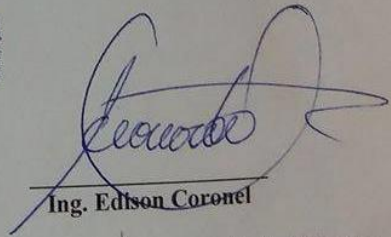
Loja, 21 de Noviembre del 2016

Ing. Milton Labanda
DIRECTOR DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN
Ing. Edison Coronel
SUBDIRECTOR DE DESARROLLO DE SOFTWARE

Con el fin de cumplir el último objetivo del proyecto de titulación denominado “*Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basa en el Modelo de Capacidad de Madurez Integrada (CMMI V1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja*”, por parte de la egresada Lady del Cisne Iñiguez Carchi se le designa como piloto de pruebas el **Módulo de Planificación y Desarrollo Institucional del “Sistema de Información Académico Administrativo Financiero (SIAAF)”** de la Universidad Nacional de Loja el mismo que cumple con los requerimientos de documentación necesaria para la realización de la validación de la propuesta.



Ing. Milton Labanda
Director de la Unidad de Telecomunicaciones e Información



Ing. Edison Coronel
Subdirector de Desarrollo de Software

Anexo 10: Elección de pruebas a realizar

PRUEBAS SOFTWARE

“Se puede definir la prueba de software como la verificación dinámica del comportamiento de un programa con un conjunto finito de casos de prueba y la comparación de su resultado con el resultado esperado” [96].

Existen distintos niveles de pruebas a menudo con diferentes intereses. Por ello se pretende realizar un revisión literaria para definir y delimitar una clasificación de pruebas; las mismas que servirán para ser ejecutadas en la etapa de control de calidad.

“Uno de los criterios más habituales para clasificar los niveles de pruebas se basa en las fases de desarrollo del ciclo de vida del software” [96]. Teniendo las siguientes:

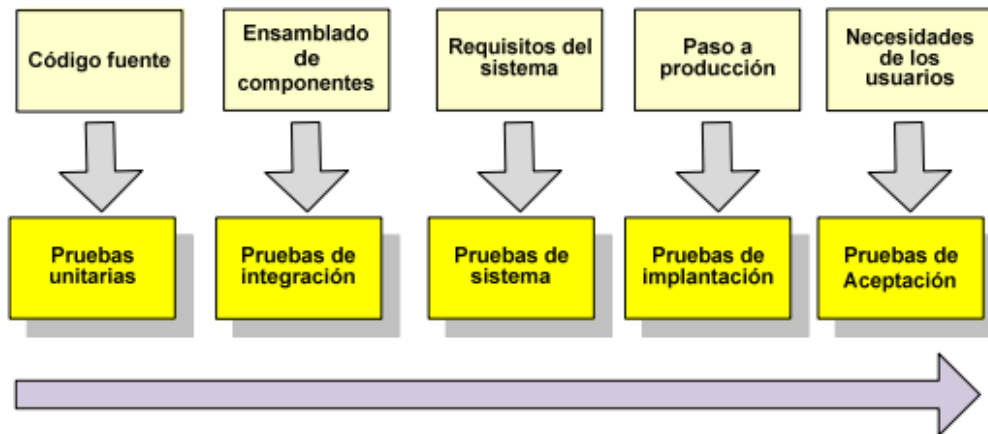


Figura 41. Niveles de pruebas [68]

TABLA XLV. CLASIFICACIÓN DE NIVELES DE PRUEBAS

Niveles de prueba	Características	Ventajas	Desventajas
Pruebas Unitarias/Componentes	<ul style="list-style-type: none"> Las pruebas de componentes (también conocidas como pruebas unitarias, modulo o programa) tiene por objeto localizar defectos en y comprobar el funcionamiento de módulos de software, programas, objetos, clases, etc., que pueden probarse por separado. [74] En general estas pruebas se llevan a cabo mediante el acceso a código objeto y cuentan con la participación del programador que escribió el código. [74] 	<ul style="list-style-type: none"> Verifica que el código no solo compila con éxito, sino que la funcionalidad está de acuerdo con las especificaciones. [85] Genera casos de prueba por componentes sin esperar que se lleve a cabo la interacción de componentes. [85] Los defectos se corrigen en el momento en que se detectan, sin gestionarlos formalmente. [74] Pueden realizarse de manera independiente del resto del sistema, sin afectarlo. [74] 	<ul style="list-style-type: none"> No se asegura que los componentes funcionen correctamente juntos. [74] Errores de interacción de componentes no pueden ser encontrados mediante la observación de componentes aislados. [74] No se puede empezar con las pruebas hasta que se disponga de componentes del sistema ya construidos. [85]
Pruebas de Integración	<ul style="list-style-type: none"> Las pruebas de integración se ocupan de las interacciones con distintas partes de un mismo sistema. [74] Este tipo de pruebas cuenta con la participación del programador que desarrollo dicho código. [85] 	<ul style="list-style-type: none"> Los probadores solo deben concentrarse en la propia integración de módulos, no en la funcionalidad de dichos módulos. [74] 	<ul style="list-style-type: none"> Los casos de prueba crecen exponencialmente a medida que el número de componentes del sistema aumenta. [97] Exigen gran cantidad de tiempo para diseñar y ejecutar los casos de prueba. [98] No se puede empezar con la planificación de las pruebas hasta que se disponga de componentes del sistema ya construidos. [99]
Pruebas de Sistema	<ul style="list-style-type: none"> Las pruebas de sistema deben de estudiar los requisitos funcionales y no funcionales del sistema y las características de calidad. [100] 	<ul style="list-style-type: none"> Verifica que el sistema software si está cumpliendo sus requisitos. [99] Permite identificar y corregir errores que pueden ser muy costosos de reparar en fases posteriores. [101] Los casos de pruebas pueden comenzar antes que el sistema esté terminado. [99] La planificación de los casos de pruebas puede comenzar tan pronto como empiece a estar disponible las especificaciones funcionales del sistema. [99] Se desarrollan en un entorno lo más parecido al entorno final de producción. [99] 	<ul style="list-style-type: none"> Tener en cuenta la tecnología empleada puesto que no se automatizan de la misma forma las pruebas a sistemas web y de escritorio. [102] Para garantizar la objetividad de las pruebas, tanto el equipo de pruebas como los participantes del negocio deben conocer el nivel de cobertura de las pruebas que se alcanzaron en las anteriores fases. [85]
Pruebas de Implantación	<ul style="list-style-type: none"> Las pruebas de implantación se encargan de comprobar que el sistema puede gestionar los volúmenes de información 	<ul style="list-style-type: none"> Asegura que el sistema se comporta de la forma prevista en el entorno de operación y que responde a todas las especificaciones dadas. [74] 	<ul style="list-style-type: none"> No se puede comenzar con la planificación de las pruebas hasta que se disponga del sistema

	requeridos, ajustándose a los tiempos de respuestas deseados. [74]		completo y se instale en su entorno de producción. [99] <ul style="list-style-type: none"> • Las pruebas de implantación cubren un rango muy amplio, que va desde la comprobación de cualquier detalle de diseño interno hasta aspectos tales como las comunicaciones. [74]
Pruebas de Aceptación	<ul style="list-style-type: none"> • El objetivo de las pruebas de aceptación es crear confianza en el sistema, partes del sistema o características, su objetivo principal no es localizar defectos sino evaluar la disposición de un sistema para su despliegue y uso. [74] 	<ul style="list-style-type: none"> • Este nivel de pruebas se lo ejecuta con el cliente, el mismo que verificara si el sistema cumple con los requerimientos solicitados. [102] • Pueden aplicarse en distintos momentos del ciclo de vida del software o sistema. [74] 	<ul style="list-style-type: none"> • Se requiere de laboratorios para la ejecución de las pruebas. [102] • No se puede comenzar con las pruebas hasta que se disponga del sistema completo y se instale en su entorno de producción. [99]

De acuerdo a la Fig. 1 se realizó el análisis detallado de cada nivel de prueba expuesto en la Tabla 1, llegando a la siguiente conclusión:

Para las pruebas Unitarias se debe contar con la participación del programador, sin asegurar que estas pruebas luego de pasar a las pruebas de Integración se tenga resultados exitosos; además las pruebas de integración son demasiado extensas ya que los casos de prueba crecen de forma exponencial a medida que el número de componentes del sistema aumentan; las pruebas de Implantación solo se las puede ejecutar cuando el sistema completo ya está en su entorno de producción y cubre un rango muy extenso desde detalles de diseño hasta aspectos como las comunicaciones; por su parte las pruebas de Aceptación se requiere la intervención del usuario y de laboratorios para su ejecución; por ultimo las pruebas de Sistema se basan en verificar los requisitos funcionales que son la base para el desarrollo del sistema estableciendo exactamente qué debe hacer dicho sistema y los no funcionales que define como debe ser el sistema y se puede empezar con los casos de prueba tan pronto como las especificaciones funcionales estén disponibles, de acuerdo a una investigación realizada los niveles de pruebas mencionadas anteriormente no son fases que se deban ejecutar rigurosamente en ese orden [85]. **Llegando a la conclusión que debido a la optimización del tiempo y recursos, el nivel de pruebas optimo a realizar es el nivel de pruebas de Sistema.**

TABLA XLVI. TIPOS DE PRUEBA DE SISTEMA [68]

Tipo de prueba de sistema	Resumen
<i>Pruebas funcionales</i>	Verifican la correcta implementación de las especificaciones funcionales del sistema.
<i>Pruebas de comunicaciones</i>	Verifican el funcionamiento de distintos subsistemas a través de sus interfaces
<i>Pruebas de rendimiento</i>	Verifican los tiempos de respuesta del sistema
<i>Pruebas de volumen</i>	Monitorizan el funcionamiento del sistema bajo distintas cargas de trabajo / datos esperados
<i>Pruebas de sobrecarga</i>	Monitorizan el funcionamiento del sistema bajo distintas cargas de trabajo / datos no esperados.
<i>Pruebas de disponibilidad de datos</i>	Verifican la respuesta del sistema ante posibles fallos
<i>Pruebas de facilidad de uso</i>	Verifican la adaptabilidad del sistema a los distintos usuarios esperados
<i>Pruebas de operación</i>	Verifican la correcta implementación de los procedimientos de operación del sistema.
<i>Pruebas de seguridad</i>	Verifican los mecanismos de control de acceso al sistema para evitar alteraciones indebidas en los datos.

Las pruebas de Sistema se clasifican en diversos tipos de pruebas los mismos que son:

A la hora de determinar que pruebas se debe realizar hay que tener en cuenta los riesgos técnicos y de seguridad, como las limitaciones del proyecto, tiempo, recursos y presupuesto. Las pruebas deben facilitar información suficiente a las partes interesadas para que éstas puedan adoptar decisiones informadas sobre el lanzamiento de un software, pasar a la siguiente fase de desarrollo o para su entrega a los clientes. [74]

De acuerdo a la optimización de tiempo y recursos disponibles con los que el GDS de la UTI cuenta actualmente, **se decide trabajar con la prueba funcional, de rendimiento y seguridad a nivel de aplicación, estas pruebas serán en estas instancias las idóneas para tener un control óptimo de calidad**; se selecciona estas pruebas porque la prueba Funcional se centran en las funcionalidades del sistema o componente además son “pruebas idóneas para realizarse en pequeñas empresas, departamentos, grupo o proyecto que cuente con pocas personas” [103] puesto que el objetivo principal de todo proyecto es tener al cliente satisfecho cumpliendo sus necesidades, de Rendimiento para determinar la velocidad con la que responde a las peticiones del usuario, un buen sistema tiene que funcionar rápido y correctamente, y de Seguridad para definir que los controles de acceso cumplen con las especificaciones que se espera y evitar intromisiones no deseadas.

Por otra parte las pruebas restantes como son de Volumen, Facilidad de uso, Comunicación, Disponibilidad de datos, Sobrecarga y Operación son pruebas costosas tanto en tiempo, recursos hardware y personal que deben estar disponibles durante todo el proceso de dichas pruebas.

Se detalla los 3 tipos de pruebas seleccionadas:

Prueba Funcional: se centran en las funciones, entradas y salidas, verifican la funcionalidad de un sistema, subsistema o componente y se las puede empezar a planificar cuanto antes se tenga los productos de trabajo. De acuerdo al ISTQB [74], la técnica para esta pruebas es la técnica de caja negra, esta técnica se centran en los resultados obtenidos con datos de entrada válidos y no válidos, sin preocuparse de lo que pase internamente en el sistema. Esta técnica tiene varios métodos para realizarla ejemplo: [74]

- *Partición de equivalencia (o clase):* se aplican datos válidos o valores que deben aceptarse y datos no validos o valores que deben rechazarse.

- *Análisis de valores límite*: el valor límite de una partición válida constituye un valor límite válido, mientras que los límites de una partición no válida constituyen valores límites no válidos.
- *Tabla de decisión*: al crear las tablas las condiciones de entrada se sentencias de manera que deban ser de tipo booleano, cada columna de la tabla corresponde a una regla comercial que define una combinación única de condiciones.
- *Transición de estados*: puede mostrarse con un diagrama de estados permitiendo ver el software en términos de sus estados, entradas y acciones, una tabla de estados muestra la relación entre estados y puede poner en manifiesto posibles transacciones no válidas.
- *Casos de uso*: se derivan de los casos de uso donde las interacciones entre los actores y el sistema produce un resultado permitiendo descubrir defectos en los flujos de proceso durante el uso real del sistema.

Por facilidad de entendimiento y por la documentación disponible a tempranas etapas en un proyecto **se trabajará con la partición de equivalencia y casos de uso**; ya que al combinarlas los casos de uso darán las interacciones entre el actor y el sistema mientras que la partición de equivalencia permite aplicar valores validos e inválidos.

- **Prueba de Seguridad a nivel de aplicación**: aseguran que el software o aplicación incorpore controles apropiados, es decir que un actor solo pueda acceder a las funciones y datos que su rol le tiene permitido.
 - *La técnica que se utiliza es identificar el tipo de usuario y las funcionalidades a las que tienen acceso dicho usuario*: se crea casos de prueba para cada tipo de usuario y se verifica que los permisos de acceso funcionen como se lo espera.

- **Prueba de rendimiento**

El objetivo es determinar si el usuario estará satisfecho con la velocidad de la aplicación.

- La técnica es utilizar una herramienta que permita simular el acceso aleatorio de usuarios y determinar tiempos de respuestas. [79]

Anexo 11: Socialización de resultados



UNIVERSIDAD NACIONAL DE LOJA

UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

Loja, 05 de enero del 2017

Ing. Milton Labanda
DIRECTOR DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN
Ing. Edison Coronel
SUBDIRECTOR DE DESARROLLO DE SOFTWARE
Ing. Jhon Calderón
SUBDIRECTOR DE REDES Y EQUIPOS INFORMÁTICOS

CERTIFICAN:

Que la egresada Lady del Cisne Iñiguez Carchi con ced. 1105333908 cuyo proyecto de titulación es *“Desarrollo de una propuesta de Mejora de Procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja”*, socializó los resultados de la ejecución del piloto de pruebas del Módulo de Planificación y Desarrollo Institucional en nuestra dependencia, habiendo cumplido satisfactoriamente con los lineamientos propuestos, mismos que serán de gran ayuda para el mejoramiento de los sistemas desarrollados en esta unidad.

Ing. Milton Labanda

**Director de la Unidad de Telecomunicaciones
e Información**

Ing. Edison Coronel

Subdirector de Desarrollo de Software

Ing. Jhon Calderón

Subdirector de Redes y Equipos Informáticos

Anexo 12: Acta entrega de documentación

ACTA DE ENTREGA

En la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja yo: Lady del Cisne Iñiguez Carchi procedí a realizar la entrega de toda la documentación generada al haber ejecutado el piloto de pruebas de mi trabajo fin de carrera titulado **“DESARROLLO DE UNA PROPUESTA DE MEJORA DE PROCESOS EN EL ÁREA DE ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE BASADA EN EL MODELO DE CAPACIDAD DE MADUREZ INTEGRADA (CMMI V1.3) PARA LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN DE LA UNIVERSIDAD NACIONAL DE LOJA”**, los documentos entregados son:

- Procesos del desarrollo de software.
- Documentos de roles y responsabilidades del grupo de PPQA y políticas.
- Plan de PPQA.
- Plantilla Checklist de Validación de Requisitos.
- Plantilla Checklist de Validación de Prototipo.
- Plantilla Checklist de Validación de Casos de Uso.
- Plantilla Informe de No Conformidades de (Validación de Requisitos, Validación de Prototipo, Validación de Casos de Uso).
- Plan de Pruebas.
- Informe de Errores.
- Informe de Aprobación.
- Documentación generada en la ejecución del plan piloto.

Para constancia de lo dicho:

ENTREGA:



Lady del Cisne Iñiguez Carchi

RECIBE:



Ing. Milton Laborda
Director de la Unidad de
Telecomunicaciones e Información



Ing. Edison Coronel
Subdirector de Desarrollo de Software



Ing. Jhon Calderón
Subdirector de Redes y Equipos Informáticos

Anexo 13: Certificado de la UTI



UNL
UNIVERSIDAD
NACIONAL
DE LOJA

Unidad de
Telecomunicaciones e
Información

Milton Leonardo Labanda Jaramillo
DIRECTOR DE LA UNIDAD DE TELECOMUNICACIONES E INFORMACIÓN

Certifica

Que la señorita **LADY DEL CISNE ÑIGUEZ CARCHI** con cédula de ciudadanía número **1105333908** egresada de la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, ha finalizado y socializado los resultados del proyecto de titulación denominado **“Desarrollo de una Propuesta de mejora de procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad de Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja”**, bajo los lineamientos y requerimientos establecidos por esta unidad administrativa

Es cuanto puedo indicar en honor a la verdad, facultando al interesado hacer uso del presente documento en lo que creyere conveniente

Loja, 18 de Enero del 2017.

Milton Labanda, Mtr

DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN



Anexo 14: Licencia Creative Commons



Desarrollo de una propuesta de Mejora de procesos en el Área de Aseguramiento de la Calidad de Software basada en el Modelo de Capacidad y Madurez Integrada (CMMI v1.3) para la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja por Lady del Cisne Iñiguez Carchi se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Internacional](#).